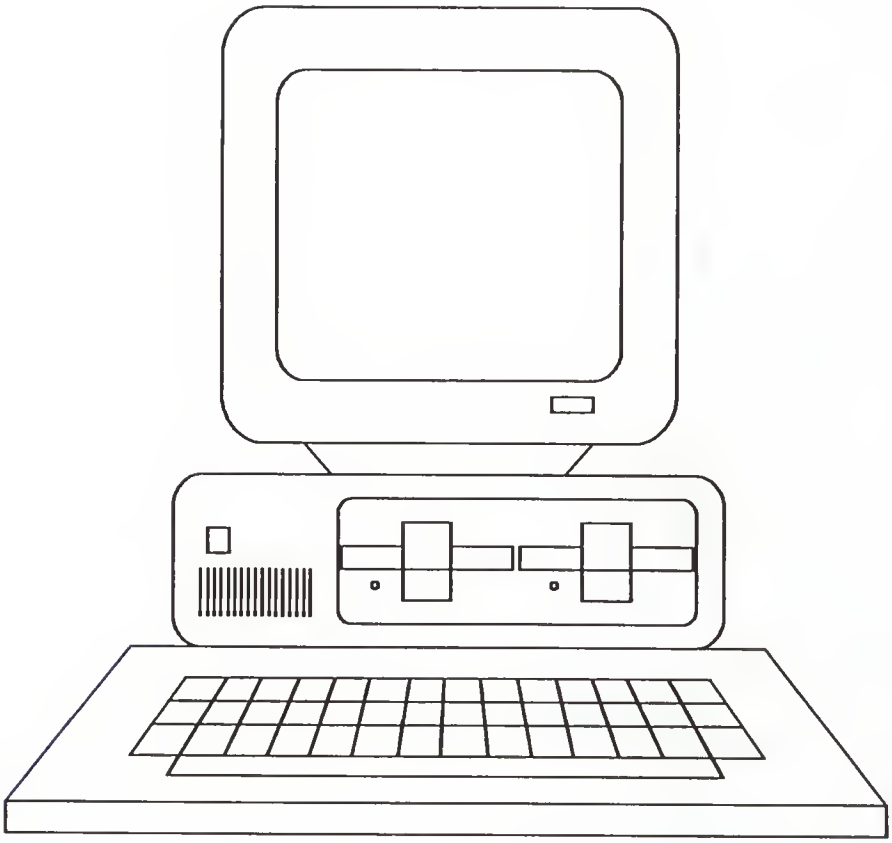
De µP Kenner

Achttiende jaargang nr. 2
Augustus 1994
86



In dit nummer o.a.:

68060 onthuld

Ervaringen met WP 6.0

CD-ROMs: een inleiding

Cursus C: laatste loodjes

DOS65: Virtual disk voorRAMkaart

Inhoudsopgave

De μP Kenner

Nummer 86, augustus 1994
Versehijnt 5 maal per jaar
Oplage: 250 stuks
Druk: FEBO Offset, Ensehede

De redactie:

Nieo de Vries
Gert van Opbroek
Geert Stappers

Eindredactie:

Nieo de Vries

Vormgeving:

Nieo de Vries

Redactieadres:

p/a Nieo de Vries
Van der Waalsstraat 46
2984 EP Ridderkerk

De μP Kenner nummer 87 versehijnt op 15 oktober 1994.

Kopijsluitingsdatum voor nummer 87 is vastgesteld op 1 oktober 1994.

Vereniging

Algemeen

Talen/Software/DOS65

Hardware/Systemen

De μP Kenner is het huisorgaan van de KIM gebruikersclub Nederland en wordt bij verschijnen gratis toegezonden aan alle leden van de club. De μP Kenner verschijnt vijf maal per jaar, in principe op de derde zaterdag van de maanden februari, april, augustus, oktober en december.

Kopij voor het blad dient bij voorkeur van de leden afkomstig te zijn. Deze kopij kan op papier, maar liever in machine-leesbare vorm opgestuurd worden aan het redactieadres. Kopij kan ook op het Bulletin Board van de vereniging gepost worden in de redactie area.
Nadere informatie kan bij het redacticadres of via het bulletin board opgevraagd worden.

Redactioneel

Het was warm de laatste maanden. Heet was zelfs niet overdreven. Om niet te spreken van zeer heet. Ik heb het over de temperatuur buiten. Ik weet niet hoe u het vond, maar ondergetekende vond het niet om te harden. De processor in mijn machine trouwens ook niet: dat is nog een antieke Intel 80486DX op 25 MHz die op eigen houtje het hete hoofd moet proberen koel te houden. Dat lukte dus met die exotische temperaturen van de laatste tijd niet zo best meer: regelmatig ging de CPU, en daarmee de gehele machine gestrekt.

De hitte heeft kennelijk ook in andere systemen toegeslagen, en wel voornamelijk in die PC's en hobbykistjes waar kopij op geproduceerd pleegt te worden. De kopijplank was deze keer zeer magertjes gevuld en dat is helaas aan dit nummer te merken. Dun derhalve. Toch is het helemaal niet moeilijk om wat regels aan de tekstverwerker toe te voegen: vertel gewoon eens een keer wat u de laatste tijd heeft meegemaakt. Een voorbeeld van zo'n verhaaltje is het artikeltje over CD-ROMs. Gewoon wat er zoals te beleven is bij de aanschaf van zo'n stukje hardware. En in dat verhaaltje zitten meteen een paar vragen, zodat er eigenlijk meteen al geldt: "wie schrijft voor het volgende nummer het vervolg?"

De eerste respons van de 1Mbyte RAMkaarten voor de DOS65 wonders is er ook al: een veranderde virtual disk driver. Komt dus in het blad. Problemen gehad met de RAMkaart? Schrijf ze op! Komt ook in het blad. Zodat we er allemaal wijzer van worden. De DOS65 werkgroep heeft trouwens nog meer plannen: er zal nu gewerkt gaan worden aan DOS65 3.00, want de direct benodigde hardware, de RAMkaart is er nu. Een vervolg wordt een samenraapkaart, die we de combo-kaart hebben gedoopt. Op dit kaartje komt een eenvoudige SCSI-interface, voor een harde schijf, reset- en NMI-generatortjes, de Elektuur Real Time Clock (welke heiden heeft een complete Eurokaart nodig voor precies één 6818 real time clock met bijbehorende logica????), een geluidsinterface-je voor de control-G's en als er nog ruimte overblijft een extra ACIA. Is de printruimte dan nog niet op, dan bedenken we een mooi lichtorgel waaraan je kunt zien waar in het geheugen de CPU aan het rondstoken is. Dat lijkt heel moeilijk, maar bestaat eenvoudig uit een slim geprogrammeerde (E)PROM die aan de adresbus hangt. De datalijnen sturen een acht-tal LEDs en dat is alles.

De bijeenkomst is NIET op 17 september maar op 24 september.

LET OP LET OP LET OP LET OP LET OP!!

Het bestuur heeft een slimme truc uitgehaald: de bijeenkomst is NIET, herhaal NIET op 17 september (de derde zaterdag van de maand) maar op 24 september. Dit heeft te maken met het feit dat op 17 september de lokaliteit in Almelo niet beschikbaar was. Op 24 september bent u allen echter meer dan welkom.

Nico de Vries

Uitnodiging voor de clubbijeenkomst

Datum: **LET OP: 24 september 1994**
Loeatie: Wijkcentrum 't Veurbrook
Jan Tooropstraat 27
7606 Almelo
Tel.: 05490 - 10353
Thema: Gedistribueerde besturing

Routebeschrijving

Vanuit het westen en het zuiden (A1/A35):

1. Aan het einde van de snelweg reehtsaf. Bij het eerstvolgende kruispunt MET VERKEERSLICHTEN linksaf, richting Wierden/ Zwolle. Bij de eerstvolgende verkeersliehten reehtdoor. Bij de volgende verkeerslichten (links BP tankstation en Opel garage Kamp) gaat u rechtsaf.

2. U rijdt nu op de Windmolenbroeksweg. Doorrijden tot over de brug, dan de eerste straat rcehts. Dit is de W. van Konijnenburgstraat. Na plm. 50 meter rechtsaf. Dit is de Tooropstraat. Met de bocht mee naar links. Na plm. 50 meter aan de reehterkant: 't Veurbrook.

Vanuit het noorden (via de N 36):

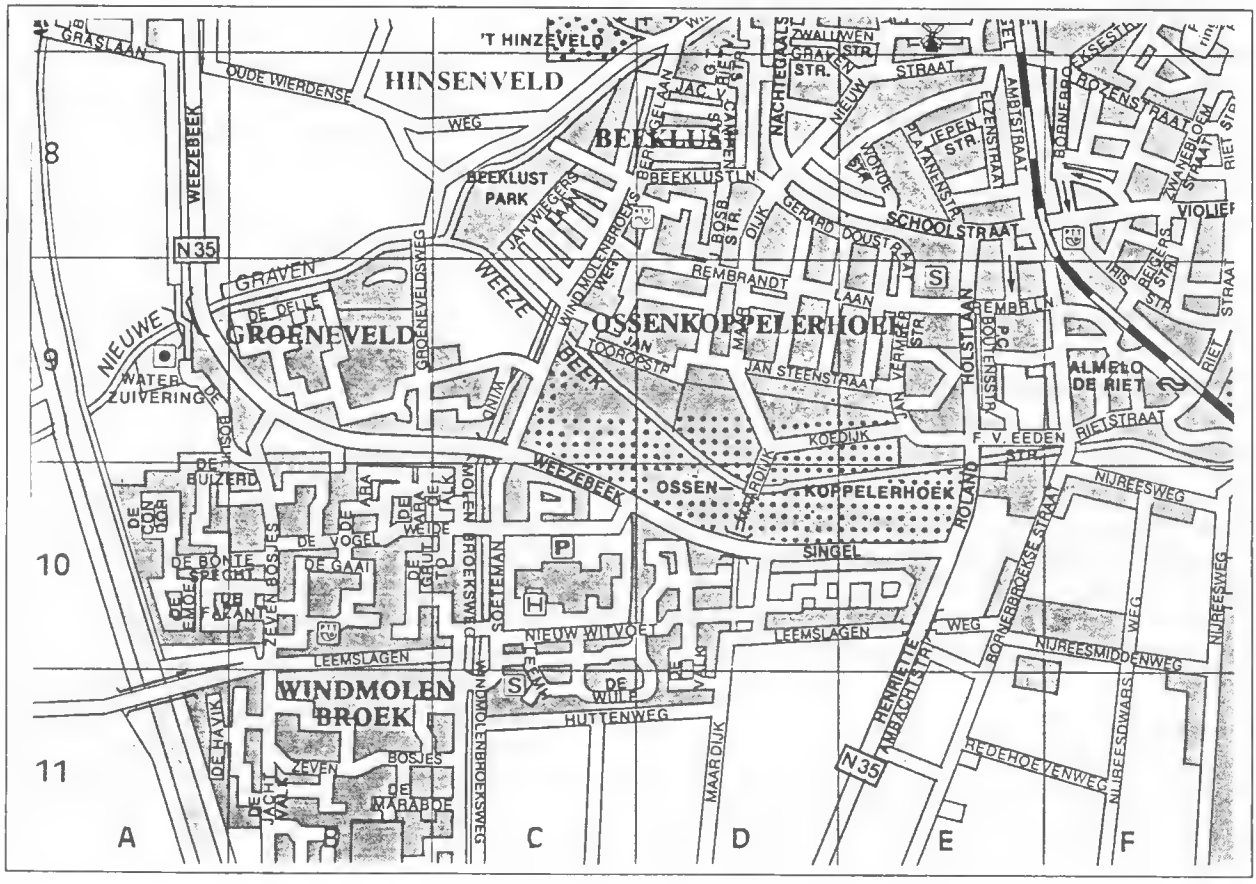
1. Bij de stoplichten rechtsaf, riehting streekziekenhuis. U bevindt zich nu op de rondweg om Almelo. Deze weg blijven volgen tot u het BP tankstation ziet bij dit kruispunt linksaf. Zie verder punt 2.

Met openbaar vervoer:

Vanaf NS-station Almelo met de stadsbus naar de wijk Molenbroek. Uitstappen bij de halte Windmolenbroeksweg. Schuin tegenover de bushalte staat een wegwijzer, daarop staat ook 't Veurbrook vermeld.

Programma:

9:30	Zaal open met koffie
10:15	Opening
10:30	Spreekbeurt door dhr. Koetsier van Controlnet over de beheersing van gedistribueerde eontrollers in een netwerk.
11:30	Forum en Markt
12:00	Lunch, eonsumpties tegen betaling Aansluitend het informele gedeelte met de mogelijkheid om andermans systemen te bewonderen en Publie Domain software uit te wisselen. U en uw systeem zijn uiteraard van harte welkom.
17:00	Sluiting.



Ervaringen met WordPerfect 6.0

Afgelopen voorjaar was het oorlog. Op het tekstverwerkerfront. Microsoft loste het eerste schot, want Nederland schakelde volgens hen over op Windows en de tekstverwerker Word. Microsoft bood een Windows-pakket en Word samen aan voor het belachelijke bedrag van ongeveer *f* 300,-. Dat Windows was geen cadeautje: Word is er niet voor DOS. En niet iedereen heeft het programmaatje Windows op zijn machine staan.

Nu is WordPerfect in Nederland marktleider op het gebied van tekstverwerkers, dus kwamen ze met een antwoord. Ze noemden het de Démarrage-aanbieding. Voor ongeveer hetzelfde bedrag boden zij de laatste versie van WordPerfect aan. De klant mocht kiezen tussen de versie voor DOS, of die voor Windows. De klant kon niet kiezen voor de taal: uitsluitend Nederlands.

Niet iedereen heeft het programmaatje Windows op zijn machine staan.

Wat het effect van beide acties geweest is weet ik niet. Wat ik wel weet dat de particulier voor een bescheiden bedrag een legale WordPerfect 6.0, compleet met support kon verkrijgen. Die particulier is waarschijnlijk de lachende derde van het oorlogsgeweld geweest, want ik ken niemand die van WordPerfect naar Word is overgestapt. Wel ken ik een hoop mensen die nu legaal thuis een WordPerfect op hun machine hebben staan. Hier en daar worden nog goedkope pakketten aangeboden.

Ik was benieuwd naar wat WordPerfect nu weer had bedacht om mij te verlokken over te stappen op een nieuwe versie. Daarom dit verhaal. Het belicht WordPerfect 6.0, gezien door de bril van een fervente 5.1 gebruiker.

Minimum machineconfiguratie

Op de doos van het pakket staat: 80286 CPU, 2 Mbyte RAM. Het pakket neemt, volledig geïnstalleerd met enkele printerdrivers, ongeveer 17 Mbyte schijfruimte in beslag (WP claimt dat een volledige installatie 14 Mbyte kost). Behalve de tekstverwerker zelf zit bij de DOS-versie ook de ASCII-editor ED in een nieuwe versie, een nieuwe WP-Shell, een complete cursus en een aantal conversie-utilities.

De overgang

Bestaande documenten in WP 4.2, 5.0 of 5.1 formaat kunnen gewoon worden opgevraagd en worden automatisch geconverteerd naar WP 6.0. Hetzelfde geldt overigens ook voor AmiPro 3.0 documenten en nog een hele berg andere tekstverwerkerformaten. Ook opslaan van documenten is in deze formaten mogelijk. De conversie is behoorlijk tijdrovend bij grote documenten.

Bestaande WP-macro's kunnen via een meegeleverd conversie-utility omgezet worden naar WP 6.0 formaat. Volgens het handboek zullen ingewikkelde macro's handmatige aanpassingen behoeven.

De bestaande woorden- en synoniemenlijsten zijn niet te converteren: deze moeten opnieuw worden aangeschaft (Kassa!). Bij WP 6.0 NL zitten alleen Nederlandse woorden- en synoniemenlijsten. De supplementaire lijsten zijn normale WP-documenten, en kunnen wel worden overgezet.

Nieuwe zaken

- WYSIWYG-presentatie. WP 6.0 kent vier presentaties van het document:
1. Tekst scherm. Lijkt sterk op het opmaakscherm van WP 5.1 en voorgaand. Displayadapter staat in tekstmode;
2. Grafisch scherm. WP probeert het document te tonen zoals dit op papier zal verschijnen, doch zonder paginakoppen en -voeten. Een pagina met 1 regel tekst wordt de getoond als die regel tekst, een page-break en het zichtbare deel van de volgende pagina;
3. Opmaakscherm. Als het grafisch scherm, maar nu echt WYSIWYG: koppen en voeten worden ook afgebeeld en een pagina met 1 regel tekst verschijnt als die ene regel in een groot wit vlak;
4. Printvoorbeeld. Dit is gelijk aan vroeger.

In het opmaak- en grafische scherm is traploos zoomen mogelijk.

- WP Corrector. Automatische correctie van veel gemaakte typefouten. Ik typ bijvoorbeeld krediet heel vaak als kredeit. De corrector maakt daar automatisch krediet van.
- WP Grammatik. Niet mee gespeeld. Stijl en grammatica controle.
- Fontbeheer. Voor mij de grootste openbaring: naast de standaard printerfonts, die voor iedere printer weer anders zijn, ondersteunt WP 6.0 ook een groot aantal printer-onafhankelijke fonts. Deze fonts zijn dan ook op vrijwel alle be-

schikbare printers te gebruiken. De printeronafhankelijke font-typen zijn onder andere:

* True-type. Overge-erfd van Windows
* Speedo. Een nieuwe naam voor iets dat al op de markt was: Bitstream Facelift Fonts
* Type 1.

Al deze fonts zijn zoals gesteld beschikbaar op vrijwel alle meegeleverde printerdefinities, inclusief PostScript printers.

- Muisbesturing. Zat al in WP 5.1, en was daar behoorlijk zinloos. In 6.0 heeft de muis meer zin, vooral in de opmaak- en grafische modi. Persoonlijk werk ik sneller met het toetsenbord.
- Knoppenbalk. Een rijtje buttons met veel gebruikte commando's erin. Kan naar keuze links, rechts, boven, onder of helemaal niet in het scherm staan. Je kunt ook je eigen knoppenbalk definiëren.
- Verbeteringen: te veel om op te noemen. Bij veel commando's zijn uitbreidingen aanwezig waaruit blijkt dat WP naar de gebruikers geluisterd heeft. Een voorbeeld: afdrukken kan nu ook met alleen de oneven of even pagina's en van achter naar voren. (Je kunt dus nu eindelijk dubbelzijdig afdrukken op een enkelzijdige printer).

Ervaringen

Als je WP 5.1 gewend bent kun je vrij snel met 6.0 aan de slag: de meeste commando's zijn hetzelfde of bijna hetzelfde. Sommige dingen zijn verplaatst. Standaard geeft 6.0 nu het Alternate keyboard, dus met Esc = Opheffen en F1 = Help. Esc = repeat en F1 = Opheffen is nu de uitzondering, en moet geïnstalleerd worden als "Toetsenbord als WP 5.1".

Ik moet hard wennen aan het Nederlands: ze kunnen bij WP nog steeds niet goed vertalen. Sommige teksten staan nog steeds in het Engels ("Sorting ToA/Index entries"), gebruiken k-spelling of zijn gewoon fout (bijvoorbeeld "Subdocumenten aanmaken?" moet zijn "Subdocumenten invoegen?").

Documenten uit WP 5.1 erven goed over, mits dezelfde printer wordt gekozen. Ik heb niets handmatig moeten bijwerken. De conversie naar WP 6.0 duurt lang (20 minuten voor een handboek van 100 pagina's).

De al aanwezige FaceLift fonts blijken ook in WP 6.0 te installeren te zijn en werken perfect. Facelift is als het ware in WP 6.0 geïntegreerd: als de fonts aan WP.DRS zijn toegevoegd, blijken alle printers ze te beheersen.

Performance

Dit was voor mij hamvraag nummer 2: hoe vlot gaat alles. WP heeft bij ondergetekende de reputatie onnodig traag te werken. Daarom is een proef gedaan met een handboek van ongeveer 100 pagina's met een index en een inhoudsopgave. Het handboek beslaat ongeveer 1 Mbyte op de harde schijf. De testmachine was een 30386 op 40 MHz, met 4 of 8 Mbyte RAM (=640 kbyte basis, 384kbyte BIOS-shadow, 3 of 7 Mbyte extended). De harde schijf is gecached met SMARTDRV, met een cache van 512 kbyte. De DOS versie is MS-DOS 3.30. Schermresolutie was 80x50 tekst en 1024x768 256 kleuren grafisch. Videokaart is een Tseng ET4000 met ge-shadowed video-BIOS.

De volgende handelingen werden losgelaten op het handboek:

- Inladen (heet in WP 6.0 plotseling "Openen")
- Naar het einde van het document springen met Home Home Pijltje neer

Versie:	5.1	5.1	6.0	6.0	6.0	6.0	6.0	6.0
Mode	-	-	Tekst	Tekst	Graf.	Graf.	Graf.	Graf.
Optie:		/R		/R		/R		/R
RAM:		4 Mb	4 Mb	4 Mb	4 Mb	4 Mb	8 Mb	8 Mb
Inladen	0'08"	0'10"	0'10"	0'25"	0'12"	0'32"	0'09"	0'09"
Naar einde	0'05"	0'08"	0'24"	1'07"	1'30"	7'02"	1'23"	0'27"
Genereren	0'31"	0'51"	8'24"	27'23"	10'10"	26'00"	3'08"	3'02"
Opslaan	0'09"	0'12"	0'32"	0'58"	0'38"	1'10"	0'19"	0'24"
Sessietijd	0'53"	1'21"	9'30"	29'53"	12'30"	34'52"	4'59"	4'02"

Fig. 1: performance vergelijking

- Inhoudsopgave en Index genereren
- Document opslaan.

De resultaten staan vermeld in figuur 1.

Hierbij geldt verder: 5.1 heeft de beschikking over 640 kbyte basisgeheugen, en 1408 kbyte EMS geheugen (5.1 werkt niet met XMS geheugen). 6.0 heeft de beschikking over 640 kbyte basisgeheugen, en 2.5 of 6.5 Mbyte XMS geheugen.

Conclusies

De volgende conclusies zijn getrokken:

1. In het meest gunstige geval is WP 6.0 viermaal langzamer dan WP 5.1, en heeft daarbij de dubbele hoeveelheid geheugen nodig.
2. Gebruik van de opstartoptie /R werkt vertragend bij (te?) weinig geheugen, maar versnellend als er voldoende geheugen beschikbaar is.
3. WP 6.0 is onacceptabel langzaam op deze machine als er te weinig geheugen aanwezig is (d.w.z. 2.5 Mbyte of minder XMS geheugen)

Samenvatting

- De WYSIWYG-presentatie in WP 6.0 zal voor aantal lieden een lang gewenst pluspunt zijn: de prijs is echter dat de performance behoorlijk slechter is.
- WP 6.0 heeft veel geheugen nodig. Bij een geheugentekort wordt alles via de harde schijf gedaan, en dit brengt het pakket bijna knarsend tot stilstand. Op een netwerk zoals wij dat hebben gaat alles nog langzamer: de lokale harde schijf is sneller dan het netwerk.
- De minimum machine voor dit pakket is een 80386/40 MHz of een 80486/25 MHz, beide met 4 of 6 Mbyte RAM. Bij 8 Mbyte begint het pakket pas lekker vlot te werken.
- Printerondersteuning en fontbeheer zijn uitmuntend;
- Onze bestaande fonts en documenten (ook AmiPro 3.0) zijn zonder meer in WP 6.0 te gebruiken.
- Nederlandse vertaling rammelt hier en daar behoorlijk.
- De Démarrage-aanbieding is werkelijk een koopje: voor *f* 300,- een krachtige WYSIWYG-tekstverwerker met een legale licentie en ondersteuning van de WP-helpdesk.

Eindconclusie en aanbevelingen

1. WP 6.0 is een vooruitgang in mogelijkheden en een achteruitgang in performance;
2. De performance is alleen acceptabel op zwaardere machines met relatief veel geheugen;
3. Voor memootjes en korte brieven werkt WP 5.1 in combinatie met het gebruikelijke printvoorbeeld eigenlijk even goed;
4. Ondergetekende heeft geen onmiddellijke behoefte om naar 6.0 over te schakelen, tenzij dit de Engelse versie wordt (geldt de aanbieding niet voor) en zijn machine wordt opgewaardeerd tot 8 Mbyte werkgeheugen.

Nico de Vries

Virtual disk op de nieuwe RAMkaart

In dit artikeltje wordt een nieuwe virtual FORMAT voorsgeteld, en wordt uit de doeken gedaan hoe de nieuwe virtual disk aan de praat te krijgen is. Het blijkt namelijk dat er ook an DOS65 zelf nog wat veranderd moet worden om de virtual disk op de nieuwe RAMkaart aan de praat te krijgen.

Wijzigingen in DOS65

Om de VRAM kaart te laten werken moet het onderstaande in DOS65 worden gewijzigd. Maak hiervoor eerst een nieuwe systeemschijf aan. Dit moet als volgt:

- Doe uw originele systeemschijf in drive 0:
- Doe een blanco diskette is drive 1:
- Formatteer de blanco de diskette;
- Kopieer alle files van de systeemdiskette naar drive 1:

In drive 1: zit nu uw nieuwe systeemdiskette.Start vervolgens de monitor op en verander de volgende zaken:

Er was:				Dat wordt:	
$CE4A	wvirt	SEC		SEC	
	rvirt	PHP		PHP	
		SEI		SEI	
		STA	$FFFD	STA	$FFED
		LDA	$FF	LDA	$FF
$CE6E		PLA		PLA	
		STA	$FF	STA	$FF
		LDA	#$02	LDA	#$0D
		STA	$FFFD	STA	$FFED
		PLP		PLP	
$CEF9		TXA		TXA	
		ADC	$CC03	ADC	$CC03
		EOR	#$0F	EOR	#$00
		BCC	$CF17	BCC	$CF17
		RTS		RTS	

Kort samengevat, u moet met de monitor de volgende bytes wijzigen:

adres	oude en nieuwe waarde	
$CE4E	FD	ED
$CE72	02	0D
$CE74	FD	ED
$CEFE	0F	00

Als dit gedaan is, de nieuwe BOOT wegschrijven met:

SAVE BOOT BF00,DFFF en vervolgens:BOOT-LINK BOOT

Doe de nieuwe systeemdiskette in drive 0: en reset het systeem. Het systeem moet nu vanaf de nieuwe drive starten. Nu kunt u de nieuwe virtual format gebruiken als 640 kByte virtual disk. Let op: Uw virtual disk wordt drive 2:. Veel plezier ermee.

Voor de rest...

Enkele opmerkingen: We hopen binnenkort de virtual format zo te verbeteren dat er 1 MByte virtual disk mogelijk is. Nuttig gebruik van VF: De meeste mensen hebben 2 drives, dan is het kopieren van een schijf ietwat onhandig, omdat de systeemschijf een drive in beslag neemt. Nu kunt u echter veel handiger kopieren: Als drive 0: de systeem schijf is, dan

- copy 1:* 2:
- floppy verwisselen
- copy 2:* 1:

Ook kunt u bijvoorbeeld uw hele systeemschijf kopieren op virtual disk, dus:

- vf
- copy 0:* 2:
- asn s2

Daarna gebruikt uw computer deze virtual disk als systeemschijf en dus starten allerlei commando's dan wel heel erg snel, en zonder allerlei herrie van uw floppen machines.

Henk Speksnijder

```
; file vformat.mac
; Format the 1 MByte RAM card as 640 kByte virtual disk.
; 5 aug. 1994 H.Speksnijder

;DOS65 i/o and special adresses
inch        equ     $C020       get character from keyboard
crlf        equ     $C02F       print CR + LF
prch        equ     $C023       print character
prhx        equ     $C038       print byte hexadecimal
prtx        equ     $C03b       print text
loupch      equ     $C041       lower to uppercase
sopt        equ     $C068       scan options
voffset     equ     $CC03       voffset for DOS65
vtracks     equ     $CC04       vtracks for DOS65
date        equ     $ABFC       4 bytes date & time

;RAM-card adresses
offset      equ     $0E         15 blocks = 60 kB ram is standard work memo
vmem        equ     $2000       memory adress used by vdisk
vdat        equ     $FFE2       adress to switch sectors of vdisk
vmap        equ     $02         original datablock at vdat

;REMARK: Changes in Dos65 BOOT:
;adres      data    explanation
;$CEFE      $00     was eor #$0f becomes eor #00
;$CE4E      $ED     was sta $FFFD becomes sta $FFED
;$CE72      $0D     was lda #02 becomes lda #$0D
;$CE74      $ED     was sta $FFFD becomes sta $FFED

            org     $200
vf          jsr     sopt        scan options
            fcc     'Y',0       only option is -Y
            bcs     90.f        if illegal option(s), report
            stx     opt         else store option
            jsr     testrom     test if you have the new EPROM
            bcs     9.f         if error, exit
            jsr     testboo     test if you hace the right BOOT
            bcs     9.f         if error, exit
            jsr     ramcard     test for card, accu=0 = NO card
            beq     91.f        if no card, report that
1           bit     opt         if '-Y' then skip askinit
            bmi     2.f
            jsr     askinit     ask if you want to initialize
            bcs     9.f         if not, exit now
2           jsr     search      else count memory size
            jsr     vdinit      initalize Vdisk
            bcs     9.f         if error, exit
            jsr     seevd       else show Vdisk size to user
9           rts                 and exit to DOS
90          jmp     vderr       incorrect options
91          jmp     vdserr      no card found
```

Fig. 1: VF.MAC

```
;TESTROM test if you have the old eprom or the new eprom.
;This routine compares a table with the eprom.
testrom     ldx     #22             get byte count
1           lda     $F033,x         get EPROM byte
            cmp     tabold,x        compare data according to old eprom
            bne     2.f             must be different
            dex                     if not, proceed
            bne     1.b             if old EPROM
            jmp     11.f            detected, report error and exit
2           ldx     #22             else get byte count
3           lda     $F033,x         get EPROM byte
            cmp     tabnew,x        compare data according to new eprom
            bne     10.f            if same
            dex                     do next byte
            bne     3.b             if done
9           clc                     clc = EPROM is OK
            rts                     and exit
10          jsr     prtx            else print string
            fcc     '\rUnknown EPROM detected in your CPU-card.'
            fcc     '\rContinue with format anyway ? Y/N: ',0
            jmp     getyn
11          jsr     prtx
            fcc     '\rOld EPROM detected in your CPU-card.'
            fcc     '\rThis should be replaced.\r',0
            sec                     sec = not ok
            rts

;TABLE with ram initialisation of old eprom
tabold      fcc     $D8               cld
            fcc     $78               sei
            fcc     $A2,$0F           ldx #$0f
            fcc     $8A             1 txa
            fcc     $49,$0F           eor #$0f
            fcc     $9D,$F0,$FF       sta $fff0
            fcc     $CA               dex
            fcc     $10,$F7           bpl 1.b
            fcc     $A9,$00           lda #$00
            fcc     $A2,$51           ldx #$51
            fcc     $9D,$FF,$E6     2 sta $ffe6
            fcc     $CA               dex
            fcc     $D0,$FA           bne 2.b

;TABLE with ram initialisation of new eprom
tabnew      fcc     $78               sei
            fcc     $D8               cld
            fcc     $A0,$00           ldy $00
            fcc     $8C,$F0,$FF     1 sty $fff0
            fcc     $A2,$0F           ldx #$0f
            fcc     $8A             2 txa
            fcc     $9D,$E0,$FF       sta $ffe0,x
            fcc     $CA               dex
            fcc     $10,$F9           bpl 2.b
            fcc     $C8               iny
```

```
                fcc     $D0,$F1             bne 1.b
                fcc     $8C,$F0,$FF         sty $fff0
                fcc     $9A                 txs

;TESTBOO check if you used the correct BOOT
testboo         lda     $CEFE               get location
                cmp     #$00                if not modified
                bne     7.f                 report error
                lda     $CE4E               get next location
                cmp     #$ED                if not modified
                bne     7.f                 report error
                lda     $CE72               get location
                cmp     #$0D                if not modified
                bne     7.f                 report error
                lda     $CE74               get location
                cmp     #$ED                if not modified
                bne     7.f                 report error
9               clc                         CLC=OK
                rts                         exit to caller
7               jsr     prtx                print string
                fcc     '\rThis is NOT the expected version of BOOT.\r'
                fcc     'Is this boot suitable for the new RAM-card ? Y/N: ',0
                jmp     getyn

;RAMCARD  test if a ramcard (new model) is in your system
; remark: this routine will not destroy any data in
; your memory or in the virtual disk.
ramcard         php                         save flags
                sei                         no interrupts, please
                lda     vmem                get data
                sta     temp                save it
                lda     #0                  write $00 in this block
                sta     vmem
                lda     #offset             switch other block in place
                sta     vdat
                lda     vmem                save data from block
                sta     temp+1
                lda     #$FF                write $FF in this block
                sta     vmem
                lda     #vmap               switch to standard block
                sta     vdat
                lda     vmem                see data in standard block
                bne     1.f                 if data=0
                lda     #offset             then card present
                sta     vdat                switch other block
                lda     temp+1              restore data there
                sta     vmem
                lda     #vmap               switch standard block
                sta     vdat
                ldx     #1
                bne     2.f
1               ldx     #0                  else NO card present
2               lda     temp                restore data
                sta     vmem
```

```
                plp                     restore flags (IRQ back on)
                txa                     accu=0=no card, accu=1=card
                rts

;SEARCH how much memory you have
; by counting the number of blocks, each block=4kB
; remark: return carry=0=ok, carry=1=no RAM card
; remark: if more than 160 tracks, use only 160 tracks.
search          lda     #offset
                sta     voffset
                sta     mut             mut = memory under test
1               jsr     is4k            test 4k, accu=0=not found
                beq     2.f
                inc     mut
                bne     1.b
2               sec
                lda     mut
                sbc     #offset
                cmp     #161
                bcc     3.f
                lda     #160            accept only < 160 tracks
3               sta     vtracks
                rts

is4k            php
                sei
                lda     mut
                sta     vdat            segment under test
                ldx     #0
                lda     #$55
                sta     vmem            test with $55
                eor     vmem
                bne     7.f
                lda     #$AA
                sta     vmem            test with $AA
                eor     vmem
                bne     7.f
                inx
7               lda     #vmap           restore old segment
                sta     vdat
                plp
                txa
                rts
61              jmp     vdserr          illegal size

vdinit          ldy     voffset
                tya
                clc
                adc     vtracks
                bcc     1.f
                bne     61.b
1               cmp     voffset         check if number of tracks is correct
                beq     61.b
                php
```

```
                sei                             no interrupts
                lda     voffset
                sta     vdat
                ldx     #0
                txa
2               sta     vmem,x                  clear sis
                sta     vmem+$100,x
                sta     vmem+$200,x             clear 1st dir
                sta     vmem+$300,x             clear 2nd dir
                dex
                bne     2.b
                ldx     #16
3               txa
                sta     vmem-1,x                sector lookup table
                dex
                bne     3.b
                lda     vtracks                 set tracks and sectors
                sta     vmem+$21                tracks
                lda     #16
                sta     vmem+$22                sectors/cilinder
                sta     vmem+$23                sectors/cilinder/side
                lda     #$FF
                sta     vmem+$24                free sector map
                sta     vmem+$25
                sta     vmem+$26
                sta     vmem+$27

                ldx     #$FF                    name
4               inx
                lda     vdname,x
                sta     vmem+$40,x
                bne     4.b
                ldx     #3                      copy date & time
5               lda     date,x
                sta     vmem+$58,x              create
                sta     vmem+$5C,x              modified
                dex
                bpl     5.b

                lda     #1
                sta     vmem+$28                allocate count=1    (sectors each bit)
                sta     vmem+$29                track shift=1
                lda     #2
                sta     vmem+$2C                data allocation track
                lda     #$F8
                ldx     vtracks                 set free bitmap
                cpx     #80+1                   check lower equal 80
                bcc     6.f                     branch if vtracks Q
                txa
                lsra
                tax
                inc     vmem+$28                allocate 2 sectors
                dec     vmem+$29                no track shift
```

```
                lda     #$04
                sta     vmem+$2F1       pointer in directory

                lda     #$FC
6               ldy     #$60
7               sta     vmem,y          set free
                lda     #$FF            ( make 16 bits/track )
                iny
                sta     vmem,y
                iny
                dex
                bne     7.b
                lda     #vmap           original segment
                sta     vdat
                plp
                clc
                rts

vdname          fcc     "Virtual disk 640k HS",0

vderr           jmp     vderr0
                fcc     $C8,$C5,$CC,$D0
vderr0          jsr     prtx
                fcc     "\rInitializes virtual disk"
                fcc     "\rSyntax: VFormat [-Y]"
                fcc     "\rOption: -Y : don't ask for permission."
                fcc     "\rAbbreviation: VF\r",0
                sec
                rts

vdserr          jsr     prtx
                fcc     "No or illegal virtual disk.\r",0
                sec
                rts

askinit         jsr     prtx
                fcc     "Are you sure to initialize the virtual disk ? Y/N: ",0

getyn           jsr     inch
                jsr     loupch
                cmp     #'Y
                beq     9.f
                jsr     crlf
                sec                     when user entered not Y then carry=1
                rts
9               jsr     crlf
                clc                     when user entered Y then carry=0
                rts

seevd           jsr     prtx
                fcc     "Virtual disk: ",0
                lda     voffset         print adress of first block
                jsr     prhx
                jsr     prtx
```

```
                fcc     "000 - ",0
                lda     voffset
                clc
                adc     vtracks         last block = voffset + vtracks
                sec
                sbc     #1
                jsr     prhx
                jsr     prtx
                fcc     "FFF\r",0       print adress of last block
                rts

endmem          res     1               1 byte number of blocks of 4 kByte
mut             res     1               1 byte memory block under test
temp            res     2               2 bytes temporary
opt             res     1               1 byte options

                end     vf
```

Ik heb interesse in de KGN en wil

☐ Lid worden van de KGN ☐ Meer informatie over de KGN

Naam : _____

Adres : _____

Postcode en Woonplaats : _____ _____

Datum : _____ Handtekening :

Dit strookje kunt u ingevuld opsturen aan het secretariaat van: KIM Gebruikersclub Nederland
Postbus 1336
7500 BH Enschede

Linux, after install (deel 1)

Wat doen de meeste mensen eignlijk met Linux heb ik eens gevraagd op een bijeenkomst. "Installeren" was het kort en bondige antwoord dat ik kreeg. Ondertussen heb ik ook Linux op mijn systeem staan. Het installeren was voor mij gedaan door een KGN clublid. Toen begon dus mij Linux, after install.

Inloggen

login vroeg het systeem aan mij. **root** en als wachtwoord alleen een return hadden ze me verteld. Er verscheen inderdaad een prompt. **whoami** had ik eens gelczen/gehoord, en jawel ik was *root*, of te wel heer en meester over het hele systeem. Toch wel een beetje beangstigend, want ik voelde me de olifant in de porseleinkast. Mezelf toen gerustgesteld dat de data op het systeem nog niet uniek was.

Meerdere terminal sessies

Eerst dacht ik dat met **altF2** de volgende terminal naar voren komt en met **altF1** de vorige terminal. Maar voor een derde terminal hielp **altF2** niet. Het blijkt dat er geen ringmechanische in zit, maar een nummering. **altF3** is de truuk om bij de derde terminal te komen.

UNIX is not an user friendly system, it requires a friendly user.

Taak overzicht

Bij een multitasking systeem zit ook een utility om een overzicht van die taken te krijgen. Bij linux heet ie **ps**. Na enig bestuderen van de uitvoer kon ik daar ook de zcs virtuele terminals terug vinden die ik gevonden had door **altF1** tot en met **altF12** uit te proberen.

Directories

Zeker weten doe ik het niet, maar ik dacht dat **ls** van *list* is. De lijst van wat er op disk staat. **ls -l** zal een bekender resultaat leveren. Veranderen van directory gebeurd gewoon met **cd directory**, *Change Directory*. Met alleen **cd** keer je weer terug naar je *home directory*. Om te weten te komen wat de huidige directory is **pwd**, *Print Working Directory*, een mogelijkheid. Maar waarschijnlijk staat het al in je prompt. Een directory omhoog doe je met **cd ..**, twee directory hoger doe je met **cd ../..**, **cd ...** had ik logischer gevonden, maarja.

RTFM

In **/usr/doc/faq** staan weer verschillende diretories, zoals bijvoorbeeld **linux**. De directory naam vertelt al wat je er kunt vinden. Read The Fantastic Manual kan bijvoorbeeld met **zless linux.faq.gz**. **zless** vind ik een prachtig commando, kijk maar eens met een andere terminal wat voor een taken er draaien (Mps -aM). Zie voor een voorbeeld van de output van dit commando figuur 1.

zcat is een **cat**, vergelijkbaar met **type** onder CP/M & MS-D0S, die tevens decomprimeert. **less** zorgt er voor dat er door heen kunt bladeren m.b.v de cursortoetsen, **ctrl-j** en **ctrl-k** zullen in ieder geval werken. Met de toets **h** verschijnt er een helpscherm, daar kun je ook lezen dat je *less* met **q** kunt verlaten.

Manual

Van de meeste zaken bestaan ook zogenaamde *manual pages*. **man man** zal dus wat over zichzelf vertellen. Wil je wat over printers weten, dan levert **man printer** een teleurstelling (**No manual entry for printer**) op. **man -k printer**, met de *k* van *keyword*, kan je dan wel verder helpen. Op je sdcherm kun je dan iets verwachten in de stijl van figuur 2.

Dan zie je ook dat er met *sections* wordt gewerkt. (De getallen tussen haakjes) De gehele indeling heb ik zo niet paraat wel weet ik dat sectie 1, de gebruikerscommando's zijn, 2 de system calls en dat 8 de system operator commando's zijn.

Editor

Bij IEDER unix systeem wordt schijnbaar **vi** meegeleverd. Een oerdegelijk stuk gereedschap. Geluk-

```
PID TTY STAT  TIME COMMAND
   54 v01 S    0:01 -sh
   55 v02 S    0:00 -sh
   56 v03 S    0:00 /etc/getty 9600 tty3
   57 v04 S    0:00 /etc/getty 9600 tty4
   58 v05 S    0:00 /etc/getty 9600 tty5
   59 v06 S    0:00 /etc/getty 9600 tty6
  181 pp0 S    0:01 -sh
  392 v01 S    0:00 -sh
  393 v01 S    0:00 zcat linux.faq.gz
  394 v01 S    0:00 less
  290 pp1 S    0:00 -bash
  395 v02 R    0:00 ps -a
```

Fig. 1: takenlijst in Linux

kig had ik "VI-the UNIX Screen Editor, A User's Guide" om er mee te leren werken. In het allereerste begin was het wel een hele klus om er aan te wennen. Maar ja *VI* is houdt zich wel aan de filosofie van UNIX. *UNIX is not an user friendly system, it requires a friendly user.* Bij jouw LINUX zal ook wel de editor **joe** zitten. Deze is wel een stuk vriendelijker. Hij heeft een on-line help en heeft veel weg van *wordstar*. **emacs** is ook beschikbaar voor LINUX, maar die zit niet bij iedere distributie.

Niewe Gebruiker

Als dan files aan gaat maken is een eigen directory een juiste plaats daarvoor. Dat je niet meer *root* bent is aanzienlijk veiliger. Een nieuwe gebruiker aanmaken is een oplossing voor deze problemen. De verschillende distributies (SLS, Slackware e.d.) hebben daar verschillende oplossingen voor. Ze heten dan **adduser** of **useradd**. Het zijn altijd *root* commando's. Het programma kan opgestart worden zonder extra opties. Na het ingeven van de nieuwe user kwam het programma zelf met zinnige default waardes. Na afloop had de file **/etc/passwd** er een regel bijgekomen. Zo een regel bestaat uit het lijstjes dat in figuur 3 is opgesomd.In **/home** was er een directory **<user_home_dir>** aangemaakt (**mkdir <user_home_dir>**). In die nieuwe directory staan kopieen van de files die in **/ect/skel** staan (**cp -R /etc/skel/.[a-z]***

/home/<user_home_dir>). Ook de eigenaar van die nieuwe directory met daar onderliggende files is aangepast. (**chown -R <user_id>:<group_ip> /home/<user_home_dir>**). Nu kun je dus ook met <user_name> inloggen. Als je in dan

<user_name>:
<gecodeerd_wachtwoord>: <user_id>:
<group_ip>: <volledige_user_name>:
<user_home_dir>: <programma_wat_opgestart_na_inloggen>.

Fig. 3: inhoud van /etc/passwd

<user_home_dir> werkt, is dat de directory die je het beste moet back-uppen.

Wachtwoord

Dat je het systeem binnen komt door bij **login root** op te geven, vindt ik ongepast. Een wachtwoord op dat machtige account dwingt je om na te denken voor je er mee gaat werken. **passwd** biedt je de mogelijkheid om het bestaande wachtwoord te wijzigingen.

Afsluiten

Een linux systeem mag je niet zomaar uitzetten. Er draaien immers nog diverse taken en er kunnen ook nog diverse files open zijn. **halt** is de opdracht die *root* moet gegeven voordat de schakelaar om mag.

Geert Stappers

1. VI-the UNIX Screen Editor, A User's Guide, August Hansen, %A Brady book, %ISBN 0_89303-928-4

man -k printer:

banner (6)	- Print large banner on printer
lp (4)	- Line printer devices
lpc (8)	- Line printer control program
lpd (8)	- Line printer daemon
lprm (1)	- Remove jobs from the line printer spooling queue
lptest (1)	- Generate lineprinter ripple pattern
pac (8)	- Printer/plotter accounting information
printcap (5)	- Printer capability data base
xdpr (1)	- Dump an X window directly to a printer

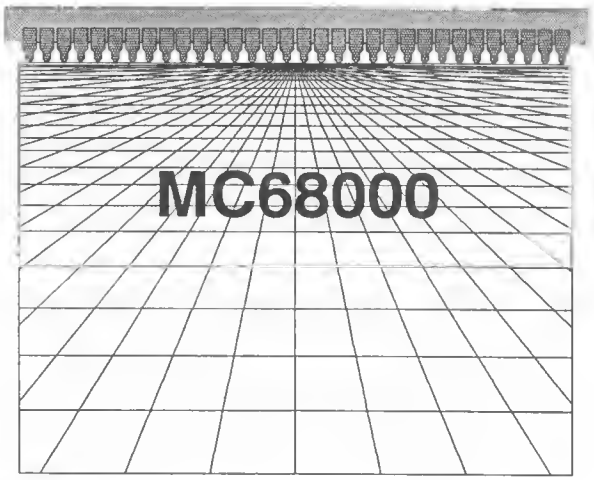
Fig. 2: outputvoorbeeld van man -k printer

De Motorola 68060: De kroon op het werk

Inleiding

Zoals waarschijnlijk wel bekend, gaat mijn voorliefde veel meer uit naar de familie van 68000-processoren van Motorola dan naar de Intel's. Ik denk dat de oorzaak hiervoor ligt in het feit dat de eerste processor die ik leerde kennen een PDP-11 was van Digital, gevolgd, door de 6502. De architectuur van de 68000-familie lijkt heel veel op die van de PDP-11 terwijl de processor bovendien goed als een opvolger van de 6502 beschouwd kan worden.

Er is sinds kort weer een nieuwe zoon van dit oude processor-geslacht op de markt. De 68060 is nu beschikbaar en het lijkt er op dat deze processor de laatste telg wordt van een processor-familie die reeds vanaf 1979 op de markt is. Aangezien niet alleen ondergetekende maar ook de KGN "iets" met 680x0 processoren heeft, is er dus alle aanleiding om deze nieuwe processor eens van dichtbij te gaan bekijken.



De 68000-familie

Er is in dit blad in het verleden al zeer veel over processoren uit de 68000-familie geschreven (ref. 1). Toch lijkt het mij zinvol nog even de diverse leden van de familie aan de lezer voor te stellen:

68008: De kleinste telg uit het roemrijke geslacht is de 68008. Deze processor heeft de beschikking over een 8 bits data-bus en een 24 bits adres-bus. Intern is de processor echter opgebouwd als een echte 32-bitter, de registers hebben een breedte van 32 bits.

68000: Dit is de stamvader van de familie. De processor heeft een 24 bits adresbus en een 16 bits databus. De 68000 loopt al weer zo'n vijftien jaar mee. Dit is de processor die te vinden is in de beroemde machines van Apple, Atari en Commodore.

68010: Ten opzichte van de 68000 is de belangrijkste uitbreiding dat de processor enkele hulpmiddelen heeft waardoor hij om kan gaan met virtuele adressen. Verder is de processor bij gelijke kloksnelheid ongeveer 30% sneller dan de 68000.

60020: Een hele belangrijke processor in de familie is de 68020. Deze processor komen we heel veel tegen in de wat duurdere 68k-machines die enkele jaren geleden ontworpen zijn. De processor heeft zowel een 32 bits brede adres-bus als een 32 bits brede data-bus. Verder heeft de processor door middel van een zogenaamde Dynamic Bus Sizing ook de mogelijkheid geheugengebieden met een breedte van 16 of 8 bits aan te spreken. Om goed samen te kunnen werken met zogenaamde coprocessors (bijvoorbeeld de 68882 mathematische coprocessor) heeft de processor hiervoor een speciaal interface. De 68020 heeft ten slotte ecn instruction cache op de chip.

68030: Deze processor is weer een uitbreiding op de 68020. De 68030 is een 68020 met behalve een instruction cache, ook nog een data cache en een geïntegreerde Memory Management Unit. Dit is de processor die ingezet wordt voor het KGN68k-project.

68040: De een vrij jonge telg uit de familie is de 68040. Behalve bijna alle eigenschappen van de 68030 en de 68020 is de 68040 ook nog een rekenwonder. De processor heeft namelijk een geïntegreerde Floating Point Unit. Helaas zijn er in de processor ook enkele dingen verdwenen die bij de 68030 juist zo handig zijn. Zo is onder andere de dynamic bus sizing van de 68020 en de 68030 hier weer verdwenen.

68060: Motorola heeft om de één of andere reden het typenummer 68050 overgeslagen. De 68060 is een verbeterde 68040. De processor is, bij dezelfde kloksnelheid

zo'n 60 à 70% sneller als de 68040. Wordt gebruik gemaakt van een compiler die geoptimaliseerde code voor de 68060 genereert, dan wordt de processor ongeveer een factor drie sneller. Verder is het ook zo dat bij een hoger typenummer de frequenties waarvoor de processor beschikbaar is hoger liggen, zodat ook daar nog een snelheidswinst uit wordt gehaald. De traagste 68040 die op de markt is draait op 25 MHz, de 68060 begint bij 50 MHz.

68881 en 68882: Alleen de twee nieuwste processoren kunnen zelf allerlei gecompliceerde berekeningen uitvoeren. Voor de iets minder getalenteerde familieleden heeft Motorola een tweetal zakrekenmachines op de markt gebracht. Dit zijn de 68881 en 68882 mathematische coprocessors. De 68882 is volledig compatibel met de 68881, hij kost alleen wat meer en is een stukje sneller. Zoals bij de 68020 al opgemerkt is, hebben de 68020 en hoger een aparte coprocessor interface waardoor de coprocessor als gelijkwaardige partner van de processor functioneert. Bij de lagere processoren kan de coprocessor het beste als een periferie chip beschouwd worden.

68851: Vanaf de 68030 hebben de processoren een interne Memory Management Unit of MMU. Voor de 68020 is er een coprocessor (de 68851) die dit klusje uitvoert. Een MMU rekent een logisch adres om in een fysiek adres. Voor de volledigheid nog even een beknopte uitleg van de MMU: Stel je voor dat een programma altijd op adres 0 begint. Als er nu meerdere programma's gelijktijdig in het geheugen aanwezig zijn, dan beginnen die uiteraard niet allemaal op fysiek adres 0. De MMU bevat een tabel waar in staat hoe de logische adressen van het lopende proces omgerekend moeten worden naar hun fysieke adres bijvoorbeeld:

Logisch	=	Fysiek
$0000 - $0FFF	=	$12000 - $12FFF
$1000 - $1FFF	=	$ 7000 - $ 7FFF
$2000 - $2FFF	=	$23000 - $23FFF

Op deze manier kun je verschillende programma's gelijktijdig in het geheugen hebben die toch allemaal dezelfde logische adressen gebruiken. Bij het opstarten van een ander programma hoef je alleen maar de tabel in de MMU te veranderen en je werkt met een ander programma. Dit is vooral erg handig bij zogenaamde Multi Tasking waarbij de processor zijn tijd verdeelt over een aantal programma's.

68230, 68681 etc. Zoals elke adellijke familie heeft de 68000 familie ook wat huis-, tuin- en keukenpersoneel. Dit zijn bijvoorbeeld een Parallel Interface/Timer en een Dual Asynchronous Receiver/Transmitter. Verder kan de familie ook zeer goed samenwerken met de periferie chips van andere dan Motorola fabrikanten.

Varianten: Behalve de bovengenoemde onderdelen zijn er nog een aantal varianten op deze chips. Dit zijn in de eerste plaats een aantal van de 68000-lijn afgeleide processoren die door Motorola zelf ontwikkeld zijn en in de tweede plaats vergelijkbare processoren van andere fabrikanten, zoals bijvoorbeeld de 68070 van Philips die veel lijkt op een uitgebreide 68000 en die onder andere in CD-spelers toegepast wordt.

Belangrijkste kenmerken van de 68060

In tegenstelling tot sommige andere processoren werkt de 68060 gewoon op 5 Volt. De gebruikte technologie voor de chip is 0,5 μm CMOS.

Dit wil zeggen dat de kleinste structuren op de chip een afmeting hebben van 0,0005 mm (hoe krijgen ze het voor elkaar....). De processor die nu op de markt komt heeft een klokfrequentie van 50 MHz maar dat zal in de toekomst nog wel wat toe gaan nemen.

Verder heeft de processor de volgende eigenschappen:

- Gescheiden pipelines voor instructies en data.
- Een zogenaamde Superscalar Pipeline voor operanden met twee afzonderlijke rekeneenheden voor Integer-bewerkingen.
- Branch Prediction Logic met behulp waarvan getracht wordt de richting van een sprongin-structie te voorspellen.
- Floating Point Unit die compatibel is met de 68881, 68882 en 68040 en die werkt volgens de IEEE-norm.
- Interne Memory Management Unit die compatibel is met de 68040.
- 8 kilobyte instructie cache en 8 kilobyte data cache.
- Een natuurlijk code compatibel met de andere leden van de 68000-familie.

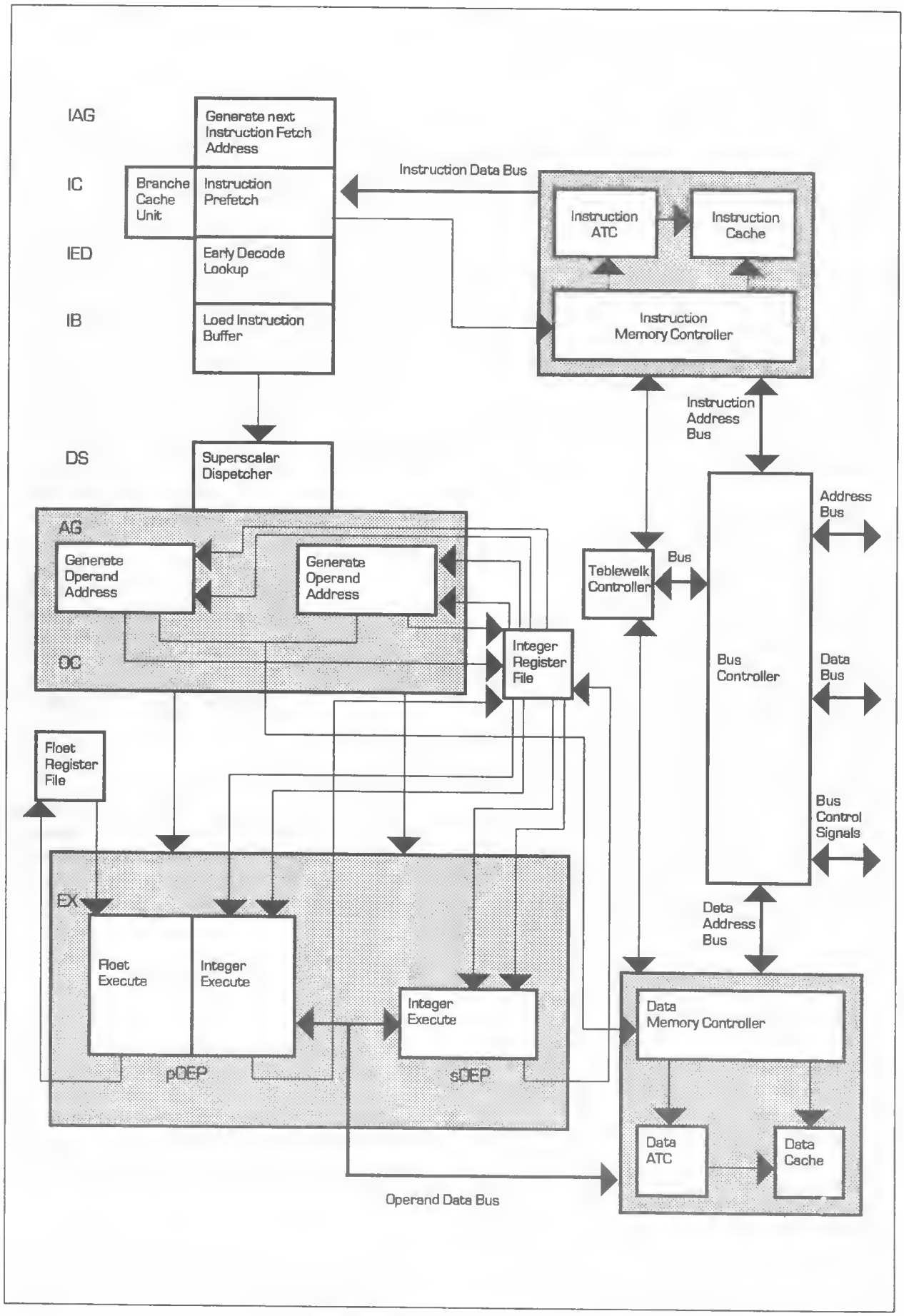


Fig. 1: interne opbouw van de 68060

Deze zaken zien er allemaal redelijk geavanceerd uit. Ondanks het feit dat de 68060 zeker geen RISC (Reduced Instruction Set Computer)-processor is, zoals bijvoorbeeld de Power PC, zijn er toch een aantal technieken ingezet die we bij de RISC-processoren ook vinden. Een belangrijk voorbeeld hiervan is wat door Motorola de Superscalar Pipeline wordt genoemd en verder verdient ook de Branch Prediction Logic wel wat nadere uitleg.

Interne opbouw

In figuur 1 is de interne opbouw van de processor schematisch weergegeven. Wat mij aan deze figuur meteen opviel is dat naast een volkomen scheiding tussen instructies en data (de zogenaamde Harvard architectuur) veel gegevenspaden dubbel zijn uitgevoerd. Laten we echter niet op de zaken vooruit lopen en gewoon bij het begin beginnen.

Rechts bovenaan vinden we de zogenaamde instruction pipeline. Dit is een soort lopende band die de opeenvolgende instructies af probeert te werken. In de eerste stap (IAG) wordt het adres van de volgend instructie bepaald. Dit adres wordt aangeboden aan de Instruction Memory Controller die vervolgens de instructie uit de instructie cache van 8 kilobyte haalt of de instructie uit het geheugen op laat halen. Nadat de instructie is ingelezen, wordt ze doorgeschoven naar de volgende stap (IC) waar de eerste stap van de decodering plaatsvindt. Ook de volgende twee stappen hebben te maken met de decodering van de instructie waarna de instructie overgedragen wordt aan de Superscalar Dispatcher (DS). De lopende band die de decodering doet wordt door Motorola IFP genoemd naar Instruction Fetch Pipeline.

Nadat de instructie gedecodeerd is, moeten de operanden opgehaald worden waarna de instructie uitgevoerd kan worden. Voor deze verwerking heeft de 68060 twee afzonderlijke verwerkings-eenheden die primairy respectievelijk secundairy Operand Execution Pipeline heten (pOEP resp. sOEP). Ik denk niet dat Motorola op de hoogte was van de nederlandse betekenis van deze woorden; nu loopt een instructie in de pOEP of in de sOEP. Goed, de Superscalar Dispatcher wijst de pipeline aan waar de instructie wordt afgehandeld. In de praktijk houdt dit in dat de processor over het algemeen met de uitvoering van twee instructies tegelijk bezig is. Elke OEP heeft de beschikking over een eigen module voor de verwerking van gehele getallen. Alleen voor de verwerking van drijvende komma getallen kan slechts een van de twee pipelines worden gebruikt.

Omdat de 68060 met de uitvoering van twee instructies tegelijk bezig is, kan het voorkomen dat deze instructies van elkaar afhankelijk zijn, bijvoorbeeld omdat ze gebruik maken van hetzelfde register. In die gevallen waarin het resultaat van instructie 1 meteen weer gebruikt wordt door instructie 2 is hier uiteraard niets aan te doen. Wel als de instructies hetzelfde register gebruiken maar niet van elkaar afhankelijk zijn. Neem als voorbeeld instructie 1 schrijft de inhoud van register D0 in het geheugen waarna het register vervolgens gevuld wordt met de inhoud van een ander geheugen adres. De 68060 lost dit op door intern meer registers te hebben (een register file) en de 16 bekende registers dynamisch in deze file toe te wijzen. Je vult dan een leeg register vanuit het geheugen voordat je D0 hebt weggeschreven en na instructie 2 noem je deze plek gewoon D0 en wordt de oorspronkelijke D0 leeg.

Oeps??!!??

Tenslotte wordt nog opgemerkt dat de OEPs in staat zijn het merendeel van de instructies in één klokcyclus uit te vocren, zoals ook een fatsoenlijke RISC-processor betaamt. Al met al is de processor in staat zo gemiddeld ruim 1 instructie per klokcyclus uit te voeren.

Sprongvoorspelling

Bij RISC-processoren zoals bijvoorbeeld de Power PC kan in een spronginstructie worden aangegeven in welke richting een sprong meestal zal gaan. Hiervan kunnen compilers optimaal gebruik maken in opdrachten zoals:

```
for (i = 0; i < 10; i+ +) .....
```

De compiler zou moeten weten dat deze lus tien keer wordt uitgevoerd zodat hij de richting van de sprong in 90% van de gevallen kent. Nu kun je in een familie niet zo maar een instructie veranderen of uitbreiden. Tenslotte heeft Motorola een naam te verliezen op het gebied van compatibiliteit zodat de spronginstructies niet veranderd konden worden.

Toch doet de 68060 aan sprongvoorspelling en wel op een zeer handige manier. Uiteraard is bij een niet-conditionele sprong (BRA = BRanch Always) bekent wat de richting van de sprong is, evenals bij aanroepen van subroutines. In die gevallen reageert

de IFP altijd op de juiste manier. Bij conditionele sprongen wordt door de processor bijgehouden of een sprong-instructie al eens uitgevoerd is en wat het volgende adres was. Dit wordt gedaan door de Branch Cache Unit. De processor gaat er dus vanuit dat als een sprong een keer in een bepaalde richting is uitgevoerd, dit de volgende keer ook zal gebeuren. Voor lussen is dit een zeer goede strategie zoals uit het regeltje C-code wel blijkt. Staat de instructie niet in de Branch Cache, dan zal de processor één van de twee mogelijke richtingen kiezen waardoor hij het toch nog in gemiddeld 50% van de gevallen bij het rechte eind zal hebben.

Nu kun je natuurlijk niet zo maar aan de uitvoering van de instructies na de sprong beginnen. Dat gebeurt ook niet. Voordat aan de uitvoering van een instructie na de sprongopdracht wordt begonnen, wordt gewacht tot er zekerheid is over de richting van de sprong en of de instructie wel uitgevoerd moet worden. Is dit niet het geval, dan blijft er niets anders over dan de hele instruction pipeline te wissen en op het andere adres opnieuw te beginnen.

De Tablewalk Controller?

Caches

Ook over de caches van de 68060 is nog wel iets te vertellen. Ik wil nu niet het hele caching mechanisme gaan behandelen; dat heb ik al gedaan bij de bespreking van de 68030 processor (μP Kenner 80). Wel wil ik wat trucjes beschrijven die de 68060 gebruikt om ook hier het onderste uit de kan te halen.

Zoals reeds vermeld, beschikt de 68060 over 8 kilobyte aan instructie cache en 8 kilobyte aan data cache. Voor de technici: het betreffen viervoudige associatieve caches met 128 regels van 16 byte. Dit betekent dat getracht zal worden in de cache in één keer 16 byte (vier longwords) in te lezen. Hiervoor heeft de processor uiteraard een zogenaamde "burst" mode waarbij deze gegevens op een snellere manier gelezen kunnen worden dan bij een normale geheugen toegang.

De data cache kan op twee manieren worden geconfigureerd. In het eerste geval worden bij schrijfoperaties op het geheugen niet alleen de cache-cellen maar ook meteen de geheugenlokaties aangepast. Dit noemen we "Write Trough Cache". In het tweede geval, de "Write Back Cache" wordt het geheugen pas later, op een geschikt moment aangepast.

Een probleem dat je in principe altijd met caches hebt, is dat de inhoud van de cache een kopie is van de inhoud van geheugenlokaties. Als er in de geheugenlokaties iets verandert, bijvoorbeeld door DMA, dan klopt de inhoud van de cache niet meer met de inhoud van het geheugen. De 68060 lost dit op door mee te luisteren op de adresbus. Komt er nu, bijvoorbeeld van de DMA controller, een adres voorbij waarvan de inhoud in het cache staat, dan wordt deze cache lokatie meteen op ongeldig gezet. Deze techniek wordt "Bus Snooping" genoemd. Ik heb als voorbeeld de DMA controller gebruikt, maar het kan ook een andere processor (bijvoorbeeld nog een 68060) zijn die het geheugen aanspreekt.

Een trucje die de 68060 in huis heeft vind je als er naar het geheugen geschreven moet worden. Het zou namelijk best voor kunnen komen dat je in het geheugen aan het lezen bent terwijl je tegelijk ook iets in het geheugen moet schrijven of dat je de inhoud van een cache regel eerst terug moet schrijven en vervolgens de regel met nieuwe informatie moet vullen. In de 68060 zijn voor deze problemen een zogenaamd "Push Buffer" en een "Write Buffer" voorzien. Het Push Buffer is een register met een afmeting van 16 byte; dus een complete cache regel. Informatie die vanuit het cache naar het geheugen toe moet worden geschreven wordt eerst in dit Push Buffer geschreven waarna de cache regel leeg is en opnieuw gevuld kan worden. Vervolgens wordt de informatie in het Push Buffer naar het geheugen geschreven.

Het Write Buffer wordt heeft een afmeting van 4 maal 4 byte. Dit buffer is er voor bedoeld om een OEP niet te hoeven stoppen als er naar het geheugen moet worden geschreven terwijl er op dat moment een leesoperatie actief is voor de tweede OEP. In dat geval wordt de informatie in het Write Buffer geschreven en kan de pipeline doorgaan met de volgende instructie. Op het moment dat er geen geheugen benadering meer actief is, wordt de informatie in het Write Buffer weggeschreven.

Overige delen

We hebben nu zo'n beetje het hele schema gehad. Er resteert nog één blokje in het schema waarover we het in het geheel niet hebben gehad: De Tablewalk Controller. Om dit stukje te beschrijven moet ik even teruggrijpen naar de behandeling van de MMU van de 68030 in μP Kenner 81. Hierin is beschreven hoe er een vertaaltabel opgebouwd wordt voor de omrekening van logische adressen naar fy-

sieke adressen. Deze tabel start in een speciaal register in de processor (de Root Pointer) en is opgebouwd in de vorm van een boom in het geheugen. Voor de omrekening van een logisch adres naar een fysiek adres wordt deze tabel afgelopen totdat de processor is aangekomen bij het blokje waar de beschrijving van het stukje geheugen staat waarna hij op zoek is. Welnu, de aansturing van het doorlopen van de tabel wordt gedaan door de Tablewalk Controller.

Een tweede zaak die misschien iets meer uitleg vraagt zijn de caches. In de figuur zijn linksboven en linksonder respectievelijk de instructie en data caches getekend. Deze caches zijn opgebouwd uit een Memory Controller voor het aansturen van het geheugen, de werkelijke cache en een Address Translation Cache. Dit laatste deel is een aparte cache waarin een deel van de tabel voor de omrekening van logische naar fysieke adressen staat. Dit houdt in dat voor het raadplegen van de omrekentabel er over het algemeen geen geheugenbenadering nodig is om dat de informatie al in de ATC staat.

Wie schakelt een processor in trace mode?

Compatibiliteit

Ondanks het feit dat Motorola compatibiliteit hoog in haar vaandel heeft staan, zijn er toch wat zaken waarin de 68060 van de 68040 afwijkt. Het betreffen voornamelijk afwijkingen in de Supervisor Mode; zaken waar je als normale programmeur niet mee te maken zult krijgen.

De 68020, 68030 en 68040 kennen drie stack pointers, een user stack pointer, een interrupt stack pointer en een master stack pointer. In de 68060 is, evenals bij de 68000 en 68010 geen master stack pointer meer aanwezig. Ook de uitgebreide trace mogelijkheden van de 68020 etc. zijn komen te vervallen. De 68060 kent slechts één trace bit waardoor het niet meer mogelijk is alleen bij subroutine sprongen en trap-instructies een trace exception te laten genereren. Ook het aparte status register voor de MMU (MMUSR) is komen te vervallen.

Ook in de Floating Point Unit zijn er een aantal wijzigingen. Bepaalde integer instructies zijn overgeheveld van de FPU naar de gewone Integer Execution eenheden en alle Floating Point sprong en Trap instructies zijn komen te vervallen. Deze instructies moeten dus door middel van een speciale exception geëmuleerd worden.

Behalve dat er dingen zijn af gegaan heeft de processor er een tweetal nieuwe registers bij gekregen. Met behulp van deze registers kan men invloed uitoefenen op hoe de processor intern moet werken, bijvoorbeeld of de pipelines of de Floating Point Unit in- of uitgeschakeld zijn.

Afsluiting

De auteur van het artikel waarop dit artikel is gebaseerd, Henrik Fisch, schrijft dat hij zich niet aan de indruk kan onttrekken dat Motorola problemen gehad heeft met de ontwikkeling van de 68060 waardoor er ten opzichte van de 68040 een aantal zaken zijn veranderd. Ik ben niet helemaal die mening toegedaan. Ik heb meer het gevoel dat een aantal zaken over het algemeen toch niet gebruikt werden (wie schakelt een processor in trace mode?) en dat ze de nieuwe ontwikkelingen in de weg stonden. Natuurlijk zullen de ontwerpers de keuze hebben moeten maken of ze zaken konden veranderen en daarmee de compatibiliteit aan te tasten maar ik denk dat ze die keuze weloverwogen hebben gemaakt met als doel de kracht van de processor zo groot mogelijk te maken.

Verder blijft de processor een volwaardige CISC (Complex Instruction Set Computer) -processor waarbij een groot aantal trucs uit de RISC processoren ook in deze processor zijn ingezet met als resultaat een processor die in staat is gemiddeld ruim 1 instructie per klokpuls af te werken; voor een CISC processor een zeer fraaie score. Het laatste argument dat voor deze processor pleit is de verregaande compatibiliteit met de voorgangers in de 680x0 familie kortom de 68060 is (voorlopig?) een fraaie afsluiting van de deze succes story.

Literatuur:

1: Gert van Opbroek: De Motorola 68030; het hart van KGN68k deel 1 t/m 6. µP Kenner 77 (augustus 1992) t/m 81 (april 1993)
2: Henrik Fisch: Motorolas MC68060: Der krönende Abschluß. MC 6 1994.

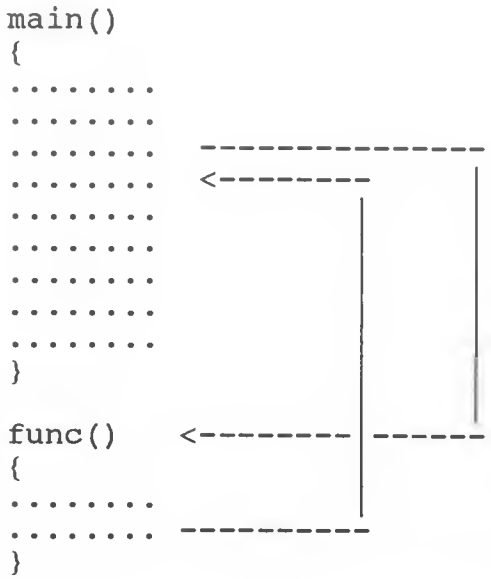
Gert van Opbroek

Laatste aflevering 'C' cursus

Als afronding van deze cursus maak je kennis met het zelf schrijven van nieuwe functies. Zoals bekend, kent 'C' zeer veel functies maar kun je ook zelf functies schrijven en toevoegen aan een library.

Deze les houden wij ons uitsluitend bezig met het schrijven van functies welke in dezelfde file zijn geschreven als de main - functie. Deze manier van toepassen wordt in andere talen vaak een 'interne sub - routine' genoemd.

Principe



Vragen...

Wij gaan eerst een aantal vragen oproepen en daarna beantwoorden:

- Hoe weet de compiler wat de hoofdfunctie is en wat de sub-functie?
- Hoe geven wij het begin en eind aan van een sub-functie?
- Hoe springen wij vanuit de main-functie naar de sub-functie?
- Op welke wijze dragen wij data over tussen hoofd- en sub-functie en terug?
- Waar declareren en assignen wij de variabelen welke in een functie worden gebruikt?
- En nog meer vragen ....

Eerste vraag

```
void naam(void);
```

Is het statement dat veelvuldig moet worden gebruikt. Een voorbeeld:

```
void lijn(void);
main()
{
 lijn();
 printf( "\xDB BOHEEMEN \xDB\n");
 lijn();
}

/* Hieronder volgt de sub-functie  */
/* Deze zet een regel op het scherm */

void lijn(void)
{
 int j;
 for ( j = 1; j < = 10; j + + )
   printf( "\xDB" );
 printf( "\n" );
}
```

De void naam(); op de eerste regel laat de compiler weten wat de naam van de sub-functie is.

Op regel 4 en regel 6 wordt de sub-functie opgeroepen. Het hoofd-programma springt hier naar de sub-functie en keert later weer terug op de eerste regel NA deze regels.

De sub-functie begint met de herhaling van het void naam () statement. Dit keer zonder een ; als afsluiting. Dat is logisch als je bedenkt dat dit ook nooit wordt gebruikt bij main().

Alles in de sub-functie tussen { en } is onderdeel van de sub-functie.

Nog enkele wetenswaardigheden:

- In dit voorbeeld wordt void gebruikt zonder uitwisseling van data tussen hoofd- en sub-functie
- void laat de compiler weten dat er een sub-functie aanwezig is. Moet daarom VOOR main() worden geplaatst.
- De declaratie van variabele 'j' vindt plaats in de sub-functie. En is daardoor alleen bekend aan de sub-functie
- xDB geeft een blokje op het scherm. Aaneengesloten blokjes vormen een regel. (in de sub-functie)
- Het geheel van de sub-functie tussen { en } wordt de 'body' van de sub-functie genoemd
- De sub-functie heeft een naam en wordt in dit voorbeeld 2x aangeroepen en uitgevoerd.

Een waarde terug naar hoofd-functie

Wij gaan nu een voorbeeld bekijken van een programma waarbij een resultaat vanuit de sub-functie wordt teruggegeven naar de hoofd-functie.

```
char getlc(void);
main()
{
 char terug;
 printf( "Type 'a' eerste selectie, 'b' voor tweede");      terug = getlc();
 switch (terug)
  {
  case 'a':
    printf ( "\nU gaf een 'a'. ");
    break;
  case 'b':
    printf ( "\nU gaf een 'b'. ");
    break;
  default:
    printf ( "\nU gaf een andere letter.");
  }
}

/* Hieronder volgt de sub-functie  */
/* Deze zet hoofd letter om in kleine letter */

char getlc(void)
{
 char ch;
 ch = getch();
 if ( ch > 64 && ch < 91 )
   ch = ch + 32;
 return ( ch );
}
```

Je ziet dat **void** nu is uitgebreid. Het eerste deel is vervangen door de een datatype. Dit datatype is hetzelfde als het datatype van de variabele waarin straks het resultaat van de sub-functie bekend is aan de hoofd-functie.

Wat ook op moet vallen zijn de aparte declaraties voor variabelen in hoofd- en sub-functie. Let op 'terug' en 'ch'.

getlc() voert sub-functie uit en plaatst het resultaat van de sub-functie in 'terug'.

Voor het omzetten van hoofd letters naar kleine letters wordt de plaats in de ASCII tabel verhoogd en daarna teruggegeven.

return() hoeft niet aan het eind van de sub-functie.

return() kan slechts EEN waarde teruggeven.

Nog een voorbeeld van waarden naar sub-functie

Nu bekijken wij een voorbeeld waarbij een samengestelde waarde wordt teruggegeven:

```
int minuut(void);
main()
{
 int minuut1, minuut2;
 printf( "Type de eerste tijd (13:32): ");
 minuut1 = minuut();
 printf( "Type de tweede tijd (13:45): ");
 minuut2 = minuut();

 printf( "Verschil is %d minuten.", minuut2 - minuut1);     }

/* Hieronder volgt de sub-functie  */

int minuut(void)
{
 int uren, min;
 scanf("%d: %d", &uren, &min);
 return (uren * 60 + min);
}
```

Ik hoef dit voorbeeld nauwelijks toe te lichten. Natuurlijk mag de berekening van het aantal minuten in de sub-functie ook apart plaatsvinden.

De functie: scanf() is niet nieuw en is in staat de inhoud van een variabele, ontvangen van de hoofd-functie, te lezen. Deze functie is bovendien in staat verschillende onderdelen te onderscheiden (in dit geval een scheiding tussen uren en minuten). Het scheidingsteken is onderdeel van de syntax van functie scanf() (in ons voorbeeld een dubbele punt maar het mag gerust een ander teken zijn).

Waarden vanuit hoofd-functie naar sub-functie

Nu bekijken wij een voorbeeld waarbij waarden vanuit de hoofd-functie worden doorgegeven naar de sub-functie. De sub-functie plaatst een boodschap op het scherm en heeft GEEN return().

```
void streep(int);
main()
{
 printf( "Piet\t");
 streep(27);
 printf( "Jan\t");
 streep(33);
 printf( "Hans\t");
 streep(14);
 printf( "Joke\t");
 streep(45);
}

/* Hieronder volgt de sub-functie  */

void streep(int score)
{
 int j;
 for ( j = 1; j < = score; j + + )
   printf("\xCD");
 printf("\n");
}
```

Opmerkingen:

- De regel voor main() (de void() regel) wordt de 'prototype' genoemd. Bij sommige 'C' compilers mag dit worden weggelaten.

- Met het oproepen van de sub-funetie wordt een waarde meegegeven. Deze waarde wordt in de sub-funetie gelezen d.m.v. void streep(int seore) en geplaatst in seore.

- Declaratie van seore vindt gelijkertijd plaats.

- De variabele "j" is alleen bekend aan de sub-funetie

- In dit voorbeeld worden vaste waarden doorgegeven aan de sub-funetie. Natuurlijk mogen deze waarden eerst worden ingetikt. Zie volgend voorbeeld.

```
void streep(int punten);
main()
{
int punten;
while (1)
 {
 printf( "Behaalde punten is: ");
 scanf("%d", &punten);
 streep(punten);
 }
}

/* Hieronder volgt de sub-functie  */

void streep(int score)
{
 int j;
 for ( j = 1; j < = score; j + + )
   printf("\xCD");
 printf("\n");
}
```

De sub-funetie is gelijk aan de sub-funetie uit het vorige voorbeeld. De hoofd-functie heeft nu een void() waarbij variabele 'punten' is ge-daelareerd. Dezelfde variabele MOET nogmaals worden ge-deelareerd in de main-funetie.

Verder is dit voorbeeld wel duidelijk lijkt mij. Let ook eens op de oneindige loop welke d.m.v. 'while(1)' is gemaakt.

Meerdere argumenten

Wij bekijken nu het gebruik van meerdere argumenten die van een hoofdroutine naar een subroutine worden doorgegeven.

```
void opperv(int,int);
main()
{
 printf( "\n Woonkamer\n");
 opperv ( 10,5);
 printf( "\n Slaapkamer1\n");
 opperv ( 8,4);
 printf( "\n Hal\n");
 opperv ( 3,2);
 printf( "\n WC\n");
 opperv ( 2,1);
```

```
 printf( "\n Slaapkamer2\n");
 opperv ( 6,4);
 printf( "\n Schuur\n");
 opperv ( 3,3);
}

/* Hieronder volgt de sub-functie  */

void opperv(int lengte, int breedte)
{
 int j, k;
 lengte /= 2;
 breedte /= 4;
 for ( j = 1; j <= breedte; j++ )
  {
  printf("\t\t");
  for ( k = 1; k <= lengte; k++ )
    printf("\xDB");
  printf("\n");
  }
}
```

Dit voorbeeld geeft op grafische wijze de oppervlakken weer - op het scherm -van de vertrekken van een woning. Omdat lengte en breedte verhouding van het scherm niet gelijk zijn is er een deling opgenomen. De manier waarop dit gebeurt is bekend uit eerdere voorbeelden met optellen of aftrekken.

Argumenten heen en terug

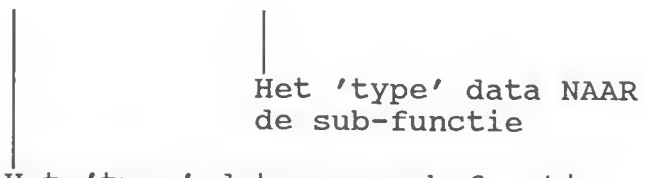
Wij bekijken nu een gecombineerd voorbeeld waarbij argumenten worden uitgewisseld tussen hoofd- en subfunctie. Dit voorbeeld berekent de oppervlakte van een bol.

```
float opperv(float);
main()
{
 float straal;
 printf( "Geef de straal van een bol: ");
 scanf ( "%f", &straal );
 printf( "De oppervlakte van de bol is %.2f", opperv(straal) );    }

/* Hieronder volgt de sub-functie  */

float opperv(float strl)
{
 return ( 4 * 3.14 * strl * strl );
}
```

Je ziet, het is allemaal erg logisch! Natuurlijk gebruiken wij 'float' omdat er wellicht een breuk kan ontstaan in de berekening; dat zou fout gaan bij het gebruik van 'int' als datatype.

Let op **void** in de prototype en in de subfunctie. Daar waar void staat (links en rechts van de naam van de subfunctie) is nu vervangen door declaraties:

```
VOID  naam (VOID)
```



Meerdere argumenten opgebouwd in een subfunctie welke teruggaan naar de hoofdfunctie moeten eerst worden samengesteld. **Return**() kan slechts een (1) waarde teruggeven. Er is echter wel een truc te bedenken: als je, in een subfunctie, meerdere keren **return**() definieert, worden meerdere waarden (net zoveel als er 'returns' zijn) teruggegeven.

Meerdere argumenten uitwisselen

Tot slot een voorbeeld met meerdere 'returns' in de subfunctie. Bovendien worden twee sub-functies gebruikt.

```
float opperv(float);
float inhoud(float);

main()
{
 float zijde;
 printf( "Geef de zijde van een kubus: ");
 scanf ( "%f", &zijde );
 printf( "De oppervlakte van de kubus %.2f", opperv(zijde) );      printf( "De inhoud van de kubus
%.2f", inhoud(zijde) );     }

/* Hieronder volgt de sub-functie   */
/* Eerst de oppervlakte          */

float opperv(float zijd)
{
 return ( 6 * zijd * zijd );
}

/* Hieronder volgt de sub-functie   */
/* Nu de inhoud                  */

float inhoud(float zijd)
{
 return ( zijd * zijd * zijd );
}
```

Dit is het einde...

Dit is het eind van de 'C' cursus voor beginners. Ik hoop dat je veel ideeen hebtopgedaan en bruist van enthousiasme om aan de slag te gaan...

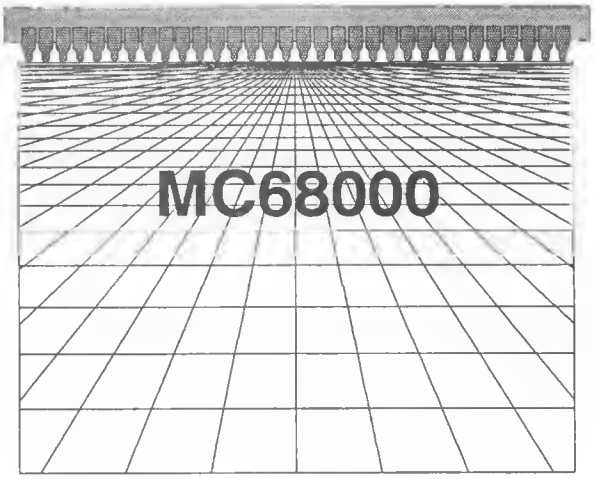
Uiteraard is dit nog maar een begin. Zoals met zo veel zaken is het ook met 'C': veel doen is de beste leermeester!

IK WENS IEDEREEN VEEL SUKSES MET "C"!

Hans van Boheemen

Voortgang KGN68k (deel 16)

Verslag over de gestaagde voortgang van het KGN68k projekt.



Hardware

De DRAM ramkaart hebben we nu ook betrouwbaar in werking. Een deel van de problemen bleek veroorzaakt te worden door te dunne bedrading in de voedingslijn op het backpanel. Nu de grote blokken af zijn, moeten we ons bezig houden met kleinere zaken die we destijds hebben laten liggen. Zoals de timer van de PI/T en de busy line van de SCSI-controller.

Software

De EPROM inhoud komt nu voortaan vanaf een Linux machine. Het is een combinatie van objects uit de KGN68k assembler en objects uit de GNU compiler/assembler verbonden door GNU-linker. Het mengen van C-code en leesbare (Motorola-syntax) assembly is nu een fluitje van een cent. De disassembler die nu ook al elke 68020 instructie kent, is zo naadloos in te voegen. De Amiga port van Linux hebben van het internet afgehaald en zijn nu bezig met het overzetten van de library routines naar onze hardware.

Printed Board Circuit

Print ontwerp is nu de bottle neck,m %Nico, het vorige stuk aub uitlichten% voor de dead-line van november 1994. Het bestuur heeft mij geadviseerd om op die plaats voor versterking te zorgen.

Marketing

Er een moment geweest dat we een commerciële kracht erbij zonden krijgen, maar helaas is dat niets geworden. Omdat in ons hart techneuten zijn blijft het bestormen van de markt nog achter wege.

Geert Stappers (04781-)41279

De KGN bijeenkomst van 26 november 1994

(in Amersfoort)

zal weer in het teken van

LINUX

staan!

We willen weer opnieuw machines in een netwerk hangen.

Wil je andere Linux zaken laten zien laat het dan weten aan het

KGN-bestuur.

CD-ROM, hebt u er al een?

U heeft ze vast al wel gezien: CD's die het niet doen in uw stereo-torentje. Vaak voorzien van aanlokkelijke afbeeldingen op het doosje. Voer voor de PC. CD-ROM heten de zilveren schijfjes die zijn bedoeld voor de computer. Ondergetekende heeft sinds kort een CD-ROM in zijn machine. Dus is er wat te vertellen.

Het begin

Ik ben net als u als ik aan iets nieuws begin: onwetend over hetgeen me te wachten staat. Daarom volg ik meestal de volgende politiek als het om nieuwe dingen gaat waarvan ik nauwelijks kan bepalen of ik het in de toekomst wel ga gebruiken of niet. Want gebruik je iets nieuws niet dan heb je misschien een hoop geld voor niets uitgegeven. Dus: eerst wachten totdat de prijzen gaan zakken. Want de "early adopters" zoals in vakstudies de haasthebbers worden genoemd, hebben weliswaar een primeur, maar betalen daar dan ook de hoofdprijs voor. Eventjes wachten is meestal lonend, want de volgende generatie van een produkt is meestal beter, goedkoper en niet langer behept met kinderziekten.

Zo ook met CD-ROM drives. Nog niet zo lang geleden waren prijzen van meer dan *f* 600,- voor een zeer simpel drive-je heel normaal. Heden ten dage niet meer: voor een anderhalf honderd pieken koop je al een drive van een gerenommeerd merk. Voor dat geld krijg je niet het allerlaatste der techniek, en evenmin het meest geavanceerde. Maar wel iets dat werkt. Dus belandde in mijn PC een drive van Panasonic. Single speed zei de verkoper erbij.

Snelheid

Want dat laatste is, zoals meestal in ons haastige PC-wereldje, van het hoogste belang. De eerste CD-ROM spelers (nog geen drives toen) waren als je goed keek verbouwde audio CD-spelers. De schijfje tolden in deze spelers met dezelfde snelheid in het rond als de CD-tjes in uw stereotoren. Ook de informatiedichtheid werd direct van de audio-broeders overgenomen. Al deze parameters leveren een niet onaardige transfersnelheid van 150 kilobyte per seconde op. Een slechte harde schijf ongeveer. Een beetje harde schijf is minimaal tweemaal sneller.

De eerste CD-ROM spelers (nog geen drives toen) waren als je goed keek verbouwde audio CD-spelers.

Sommige mensen vonden die 150 kbyte per seconde te langzaam. Dus verhoogde fabrikant Panasonic de draaisnelheid van de drive met een factor twee en doopte dat double speed. De oude drives waren vanaf dat moment dus automatisch single speed geworden. Inmiddels is double speed zo ongeveer de standaard geworden, en is single speed voor lieden die geen zin hebben om veel geld uit te geven, maar wel een hoop geduld hebben.

Triple speed (dus drievoudige transferrate en dito draaisnelheid) en zelfs quadruple speed (alles keer vier alstublieft) wordt ook al aangeboden. Maar hiervoor moet nog wel stevig in de beurs getast worden.

Meer dan 1 disc, trays en caddy's

De drive in mijn PC heeft een zogenaamde caddy. Dat is een soort cassette waar je de CD-ROM in moet doen. De cassette gaat met CD en al in de drive. Het kan ook simpeler: met een tray zoals bij de meeste audio CD-speler. Trays zitten in de duurdere spelers, caddy's in de oudere en goedkopere.

Sommige fabrikanten verraden graag hun afkomst. Pioneer, bekend van audio apparatuur van redelijke tot uitstekende kwaliteit vond voor de stereofreak met tegenzin tegen lopen de CD-wisselaar uit. Dat apparaat bevatte een magazijn waarin zes CD's konden worden opgeborgen. U raadt het al: er bestaat van Pioneer ook een double CR-ROM wisselaar.

Interfacing

CD-ROM drives bezitten, afhankelijk van het fabrikaat een steeds verschillende parallel-aansluiting. Afhankelijk van de herkomst kan dat een Mitsumi-, Sony- of Panasonic aansluiting zijn. De interface stelt verder niet zo veel voor, want het kaartje dat bij mijn drive zat bevat meer connectors dan IC's. En de IC's op dat kaartje zijn allemaal van die simpele zaken als een PAL en wat buffers. Dat komt omdat het werkelijke werk niet door de interfacekaart wordt gedaan, maar door het moederbord van uw eigen PC. Want daar zitten een paar prachtige zaken in die bijna nooit worden gebruikt. Zoals DMA-controllers. U raadt het al: de CD-ROM werkt onder DMA. Met ook een interruptje erbij. Het kaartje van mijn drive gebruikt het populaire I/O-adres 0300h. De bijgeleverde documentatie vermeld geluk-

kig ook, dat het kaartje verhuist kan worden naar 0310h, 0320h en 0330h.

Een andere, duurdere manier van interfacing bestaat uit een geluidskaart. Veel van die kaarten bezitten ook een aansluiting van CD-ROM's. Daarbij is het even uitkijken: niet alle kaarten ondersteunen alle soorten aansluitingen. Zo is meneer Sound Blaster getrouwd met drives van Panasonic.

De laatste manier van interfacen is via de vertrouwde SCSI-bus. Dit vereist uiteraard een drive met een SCSI-interface.

In het operating system

Hoe knoop je nou zo'n CD-ROM drive aan je PC? De meegeleverde instructies waren in mijn geval bepaald spaarzaam. Als eerste moet je een driver in de CONFIG.SYS opnemen met een DEVICE= statement. Sommige drivers accepteren een massa verschillende parameters, en andere maar één: de naam van de drive. Want onder MS-DOS heeft de CD-ROM drive een naam. Hij wordt opgegeven met de /D-parameter.

Mits er een programmaatje in de PC wordt gestart, kan de CD-ROM drive ook audio-CD's draaien.

De tweede stap komt in de AUTOEXEC.BAT en betreft het aanroepen van de driver MSCDEX.EXE. Ook hier wordt met een /D-parameter een naam opgegeven. MSCDEX koppelt de naam nu aan een driveletter. Welke letter dat wordt kan ook worden beïnvloed. Vanaf dat moment ziet de CD-ROM drive eruit als een enorme harde schijf waarop je echter alleen kunt lezen.

Onder OS/2 gaat het eenvoudiger: driver opgeven in CONFIG.SYS en klaar is CD. Onder OS/2 kun je de driveletter niet beinvloeden.

Wat kun je er nu mee?

Veel en bijna niets. De CD-ROM is maximaal 630 Mbyte groot. Hij leent zich dus uitstekend voor het overbrengen en opslaan van grote hoeveelheden data of data die van nature veel ruimte vereist zoals plaatjes en geluidsfragmenten. Plaatjes zijn er in de CD-ROM wereld te over. Vooral als u van afbeeldingen met een zeer directe link met moeder Natuur houdt zijn de mogelijkheden zo ongeveer onbegrensd. Dit is wat mij betreft de afdeling bijna niets.

Zinniger worden opzoek-CD's. Zo zijn er film-encyclopedies, gewone encyclopedies, programma bibliotheken en andere verzamelwerken, zoals lijsten van medicijnen of allerlei gegevens over één onderwerp zoals dinosauriërs of de JFK-moord.

Mits er een programmaatje in de PC wordt gestart, kan de CD-ROM drive ook audio-CD's draaien. De meeste drives bezitten een hoofdtelefoon aansluiting, en ook een volumeregelaar.

Een andere stroom CD's wordt gevormd door omvangrijke softwarepakketten. Zo is het uitgebreide operating system OS/2 verkrijgbaar op CD-ROM. Volgens gerucht zal Windows NT alleen maar op CD-ROM uitkomen. Ook andere volumineuze pakketten zitten op CD-ROM. Ik zag bijvoorbeeld Corel-Draw en verschillende pakketten die fonts bevatten.

Kwantiteit is ook iets dat hoort bij shareware. Dus zijn er ook een hele rij goedkope CD-ROMs (vanaf *f* 20,-) afgeladen met shareware. Zo zat gratis bij mijn drive een CD-ROM met een complete Linux erop, naast spelletjes en een berg OS/2 utilities.

Kortom: er is voor elk wat wils.

Vragen

Zoals aan het begin gesteld weet ik nog langs alles niet over CD-ROMs. Bijvoorbeeld: wanneer is een CD-ROM drive geschikt voor het afspelen van Photo-CD's? Wordt dat alleen bepaald door de sturende software, of worden er ook eisen aan de drive gesteld? Of: wat is multi-session? Wat kun je daar mee? Wie schrijft hier eens iets over?

Nico de Vries

UNIX toegangsbeveiliging

Computersystemen, welke gebruikt worden door meerdere personen, maken bijna altijd gebruik van een toegangsbeveiliging. Om toegang tot deze computers te krijgen maken we gebruik van een login naam (vaak bestaand uit een cryptische afkorting van voor en achternaam) en een geheime toegangscode (password). Dit password is meestal alleen bekend bij de persoon die inlogt, maar kan in veel gevallen ook door de systeembeheerder worden bepaald.

Onder UNIX worden de usernamen en password normaal gesproken opgeslagen in de file "/etc/passwd". Deze file bevat voor elke gebruiker een regel met de volgende informatie:

- login naam
- versleuteld password
- nummer voor deze gebruiker
- nummer voor de groep van de gebruiker
- veld voor extra informatie
- werkdirectory
- shell programma

We maken gebruik van een login naam (vaak bestaand uit een cryptische afkorting van voor en achternaam)

Login naam:

Dit veld is maximaal 8 karakters lang, en bevat meestal een afkorting van de werkelijke naam van de gebruiker. Voor mij zal bijvoorbeeld de loginnaam "jve" gebruikt kunnen worden.

Versleuteld password:

Het password kan door de gebruiker zelf worden gewijzigd met het programma "passwd". Als een nieuwe gebruiker wordt aangemaakt gebeurt dat vaak met een leeg password, waarna de gebruiker dit zo snel mogelijk moet veranderen. Het password programma vraagt altijd eerst naar het vorige password, om te voorkomen dat een onbevoegde het password kan wijzigen. Hierna wordt het nieuwe password gevraagd. Het passwd programma heeft meestal een aantal regels voor het aeeepteren van passwords om te voorkomen dat te eenvoudige passwords worden gebruikt. Daarna wordt het password nogmaals gevraagd, om te verifieren of dit werkelijk het bedoelde nieuwe password is. Is alles in orde dan wordt het password versleuteld opgeslagen in de file "/etc/passwd". Wanneer dit veld met een texteditor gewijzigd wordt zal het werkelijke passwd niet meer bruikbaar zijn. Als een gebruiker inlogt zal het inlogprogramma het ingetoetste password ook gaan versleutelen en vergelijken met de juiste regel in de "/etc/passwd" file. Komen de versleutelde velden overeen dan is het password juist ingevoerd en krijgt de gebruiker toegang tot het systeem.

Nummer voor deze gebruiker

Elke gebruiker heeft in het systeem een uniek nummer. Dit nummer wordt intern gebruikt om bijvoorbeeld te bepalen van wie een bepaald process of een bepaalde file is. In combinatie met ingestelde permissies wordt hiermee dan vastgelegd wat een gebruiker wel en niet mag doen. Een nummer (0) is gereserveerd voor de superuser (meestal loginnaam root). Deze gebruiker heeft veel meer rechten in het systeem en wordt meestal alleen door de systeembeheerder gebruikt.

Nummer voor de groep van de gebruiker

Gebruikers worden ingedeeld in groepen. Dit wordt gedaan om het mogelijk te maken dat een bepaalde groep mensen met elkaars files kan werken. Door de juiste instellingen is het mogelijk dat een collega van mij bepaalde documenten kan raadplegen die ik voor hem heb vrijgegeven. Ik zet de permissies dan zodanig dat hij alleen kan lezen en dus niet schrijven. Groepen hebben ook namen die in de file "/etc/group" staan. Voor elke groep staat er een regel in deze file met de volgende velden:

- naam van de groep
- versleuteld password
- nummer van de groep
- gebruikers die toegang hebben tot deze groep

Het password in het tweede veld kan gebruikt worden om van de huidige groep over te stappen in een andere groep zodat je weer andere rechten op bepaalde files hebt. Dus je behoort standaard tot een bepaalde groep, maar je kan tijdelijk overstappen naar een andere groep. Ik heb dit nog niet veel gebruikt, en kan er dus niet teveel over vertellen.

Veld voor extra informatie

Hierin kan informatie worden geplaatst als werkelijke naam, afdeling en telefoonnummer. Dit veld wordt door programma's als finger gebruikt voor het

maken van nette overzichten, maar dit veld moet dan wel op een bepaalde manier worden ingevuld.

Werkdirectory

Hierin staat de directory waarin de gebruiker na inloggen begint. In deze directory staat meestal een bestand genaamd ".profile" (lijkt op de AUTOEXEC.BAT van MS-DOS) waarin allerlei instellingen gemaakt of programma's opgestart kunnen worden. Na inloggen wordt dit UNIX script uitgevoerd.

Shell programma

Dit is het programma dat gestart wordt om de commando's van de gebruiker aan de nemen (vergelijk met COMMAND.COM van MS-DOS). Dit programma verwerkt ook de .profile uit de vorige alinia. Voorbeelden zijn "/bin/sh", "/bin/bash"

De toegangsbeveiling zoals hierboven beschreven wordt vaak niet als voldoende beschouwd. Daarom zijn er door verschillende UNIX leveranciers allerlei extra's toegevoegd, die vaak onderling verschillen. Het hangt van de gevoeligheid van de informatie en de toegangsmogelijkheden tot het systeem (een computer zonder netwerk in een afgesloten ruimte met een terminal is natuurlijk eenvoudiger te controleren dan een systeem aangesloten op internet of een stelsel van netwerken bij verschillende filialen door heel europa) af of en voor welke extra beveiligen er gekozen moet worden. Maar het zal altijd moeilijk blijven een systeem volledig te beschermen tegen niet gewenst danwel crimineel gebruik.

Jan Veninga

Van de voorzitter

Het computer seizoen 94/95 is weer begonnen.

Het belooft een spannend seizoen te worden. De verschillende vragen die er liggen zijn mijn inziens van levensbelang voor de KGN. Komt KGN68k op tijd klaar? Kunnen we dan genoeg lawaai maken om de machine en/of de technologie op de markt te brengen? Gaan de ontwikkelingen met DOS65 verder? Staat er iemand op met een goed idee? Kunnen opnieuw drie bestuursleden de KGN draaiende houden?

Het belooft een spannend seizoen te worden.

Een antwoord van stuur*lui* aan de wal is dan "De tijd zal het leren". Daar hebben ze opzich gelijk in en verder zullen ze zich waarschijnlijk niet druk maken. Wat ik weet is dat als er niets gebeurd, het moeilijk zal zijn om nog aan een volgend seizoen te beginnen.

Gelukkig zijn er reeds bestaande werkgroepen die geestdriftig bezig zijn onze hobby. De resultaten daarvan zijn zaken waar we als club trots op kunnen zijn. Bescheidenheid is best een goede deugd, maar teveel is ook weer niet goed. Om te vertellen hoe goed iets is, is weer een andere kunst. Als meest geschikte plaats om in aanraking te komen met de buitenwereld hebben we de HCC-dagen in gedachten. Daar willen we dan lawaai maken door onze aanwezigheid. De prijs die de organisatie vraagt is echter behoorlijk prijzig. In een brief naar hun wordt gevraagd naar de mogelijkheid voor een prijsonderhandeling. Daar zal dan de beursdeelneming van afhangen.

Met meer leden neemt ook de kans toe op iemand met nog een goed idee. Dat hij/zij er ook mee naar voren durft te komen hoop ik van wel. Zelf vind ik het wel leuk om zaken aan andere mensen te tonen die weten hoeveel werk je verzet hebt. Dat is trouwens ook een van de vele leuke kanten van de KGN. Voor een club zoals de KGN is dan ook zeker plaats. De huidige (bestuurs-)leden en de nieuwe (bestuurs-)leden zullen dat zeker beseffen.

Gegroet, jullie voorzitter,

Geert Stappers

KGN promotie team

Jullie als lid van de KGN mogen deel uitmaken van het KGN promotie team.

Het is gelukkig geen full-time job, want we hebben het waarschijnlijk toch wel druk genoeg. We zoeken een groep mensen die bereid zijn eventueel een stand te bemannen tijdens een beurs of ander evenement. Als je nu ja zegt betekent dat niet dat je straks nog aan die verplichting vastzit. Pas als de KGN naar een namelijk in het aansluiten van de te demonstreren computer(toepassing). Tijdens de beurs zelf promotiemateriaal uitdelen, eventueel wat vertellen. In overleg met andere KGN promotie team leden wie er bij de stand blijft en wie er even over de beurs wandelt. Aan het einde van de dag de zaken weer netjes afbreken.

Hoe wordt je KGN promotie team lid?

Een briefje, bericht of telefoontje naar Geert Stappers!

Informatie

De μP Kenner (De mieroprocessor Kenner) is een uitgave van de KIM gebruikerselub Nederland. Deze vereniging is volledig onafhankelijk, is statutair opgericht op 22 juni 1978 en ingesehreven bij de Kamer van Koophandel en Fabrieken voor Hollands Noorderkwartier te Alkmaar, onder nummer 634305. Het gironummer van de vereniging is 3757649.

De doelstellingen van de vereniging zijn sinds 1 januari 1989 als volgt geformuleerd:

- Het vergaren en verspreiden van kennis over componenten van mierocomputers, de miero-eomputers zelf en de bijbehorende systeemsoftware.
- Het stimuleren en ondersteunen van het gebruik van miero-eomputers in de meer teehnisehe toepassingen.

Om deze doelstellingen zo goed mogelijk in te vullen, wordt onder andere 5 maal per jaar de μP Kenner uitgegeven. Verder worden er door het bestuur per jaar 5 landelijke bijeenkomsten georganiseerd, beheert het bestuur een Bulletin Board en wordt er een softwarebibliotheek en een teehnisch forum voor de diverse systemen in stand gehouden.

Landelijke bijeenkomsten:

Deze worden gehouden op bij voorkeur de derde zaterdag van de maanden januari, maart, mei, september en november. De exaete plaats en datum worden steeds in de μP Kenner bekend gemaakt in de rubriek Uitnodiging.

Bulletin Board:

Voor het uitwisselen van mededelingen, het stellen en beantwoorden van vragen en de verspreiding van software wordt er voor de vereniging een Bulletin Board (BBS) besehikbaar gesteld.
De telefoonnummers zijn: 053-328506, 053-303902 of 053-327457.
Berichten sturen (Mail) via Internet naar het BBS is tegenwoordig ook mogelijk.
Het internet adres van het BBS is:: ultima.iaf.nl

Software Bibliotheek en Technisch Forum:

Voor het beheer van de Software Bibliotheek en teehnisehe ondersteuning streeft het bestuur ernaar zgn. systeemeoordinators te benoemen. Van tijd tot tijd zal in de μP Kenner een overzieht gepubliceerd worden. Dit overzieht staat ook op het BBS.

Correspondentie adres

Alle correspondentie betreffende verenigingszaken kan gestuurd worden aan:

KIM Gebruikerselub Nederland
Postbus 1336
7500 BH Ensehede

Het Bestuur

Het bestuur van de vereniging wordt gevormd door een dagelijks bestuur bestaande uit een voorzitter, een seeretaris en een penningmeester en een viertal gewone leden.

Geert Stappers (Voorzitter, KGN/68k coordinator)
Engelseweg 7
5825 BT Overloon
Telefoon 04781-41279
Internet: stappers@knoware.nl

Jaeques H.G.M. Banser (penningmeester)
Haaksbergerstraat 199
7513 EM Ensehede
Telefoon 053-324137
Internet: jaeques@ultima.iaf.nl

Gert van Opbroek (seeretaris)
Del Del 16
5071 TT Udenhout
Telefoon 04241-3795

Jan Veninga
Klimopstraat 51
7601 SJ Almelo
Telefoon 0546-827910

Henk Speksnijder
Albert Cuijpstraat 43
2902 GA Capelle aan den IJssel
Telefoon 010-4586879

Nieo de Vries (redaetie μP Kenner)
Van der Waalsstraat 46
2984 EP Ridderkerk
Telefoon 01804-29207

Ereleden:

Naast het bestuur zijn er een aantal ereleden, die zieh in het verleden bijzonder verdienstelijk voor de elub hebben gemaakt:

Erevoorzitter:
Siep de Vries

Ereleden:
Mevr. H. de Vries-van der Winden
Anton Müller
Rinus Vleeseh Dubois